**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**
**імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформаційних систем та технологій**

«На правах рукопису»
УДК 004.75

До захисту допущено:
В. о. завідувача кафедри
_____ Олександр РОЛІК
«___»_____2021 р.

# Магістерська дисертація

**на здобуття ступеня магістра**
**за освітньо-професійною програмою «Інформаційне забезпечення робототехнічних систем»**
**зі спеціальності 126 «Інформаційні системи та технології»**
**на тему: «Підсистема моделювання маршрутів руху мобільного транспортного робота»**

Виконала:
студентка VI курсу, групи ІК-01мп
Кривич Ольга Сергіївна _____

Керівник:
доцент кафедри ІСТ, к.т.н., доц.,
Резніков Сергій Анатолійович _____

Рецензент:
доцент кафедри ІПІ, к.т.н., доц.,
Баклан Ігор Всеволодович _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студентка _____

Київ – 2021 року

**Національний технічний університет України**
**«Київський політехнічний інститут імені Ігоря Сікорського»**
**Факультет інформатики та обчислювальної техніки**
**Кафедра інформаційних систем та технологій**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри

_____ Олександр РОЛІК

«___»_____2021р.

**ЗАВДАННЯ**
**на магістерську дисертацію студенту**

**Кривич Ользі Сергіївні**

1. Тема дисертації «Підсистема моделювання маршрутів руху мобільного транспортного робота», науковий керівник дисертації Резніков Сергій Анатолійович, к.т.н., доцент, затверджені наказом по університету від «27» 10 2021 р. № 3587-с

2. Термін подання студентом дисертації «13» грудня 2021 р.

3. Об'єкт дослідження – рух мобільних роботів по заданій карті з виконанням типових операцій

4. Предмет дослідження – алгоритми пошуку раціонального маршруту виконання типових операцій (завантаження, розвантаження та очікування) роботом в промислових приміщеннях по заданій карті

5. Перелік завдань, які потрібно розробити – аналіз існуючих алгоритмів пошуку найкоротшого шляху, вибір технологій для реалізації, проектування підсистеми пошуку оптимального маршруту (плагіна), розробка та візуалізація роботи плагіна у 3D середовищі.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу – 8 плакатів

7. Орієнтовний перелік публікацій – одна публікація

8. Консультанти розділів дисертації

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| | | | |
| | | | |

9.Дата видачі завдання «1» вересня 2021 р.

Календарний план

| № з/п | Назва етапів виконання магістерської дисертації | Термін виконання | Примітка |
|---|---|---|---|
| 1 | Огляд існуючих алгоритмів для пошуку оптимального шляху | 01.09.2021 – 07.09.2021 | |
| 2 | Огляд алгоритмів розбиття мапи для побудови графі | 08.09.2021 – 14.09.2021 | |
| 3 | Налаштування середовищ для розробки та моделювання | 15.09.2021 – 21.09.2021 | |
| 4 | Розбиття мапи на граф шляхів | 22.09.2021 – 28.09.2021 | |
| 5 | Розробка побудови шляху робота по графу шляхів на мапі | 29.09.2021 – 04.10.2021 | |
| 6 | Програмна реалізація плагіну | 05.10.2021 – 20.10.2021 | |
| 7 | Тестування та візуалізація плагіну у 3D середовищі | 21.10.2021 – 10.11.2021 | |
| 8 | Оформлення пояснювальної записки | 11.11.2021 – 23.11.2021 | |
| 9 | Подання дисертації на попередній захист | 24.11.2021 | |
| 10 | Подання дисертації на захист | | |

Студент Ольга КРИВИЧ

Науковий керівник Сергій РЕЗНІКОВ

# РЕФЕРАТ

Розмір пояснювальної записки – 102 аркушів, містить 44 ілюстрацій, 24 таблиць, 8 додатків.

**Актуальність теми.** У роботі розглянуто проблему пошуку оптимального шляху для пересування чотириколісного мобільного робота в умовах технологічного приміщення з виконанням типових операцій розвантаження, завантаження та очікування. Обґрунтовано основні недоліки сучасних рішень, а саме потреба в повному переобладнанні приміщень для застосування подібного роду систем. Виявлено потребу в спрощенні процесу розгортання системи керування рухом та потребу в розробці методології чи критеріїв для оцінки вартості того чи іншого алгоритму пошуку шляху, запропоновано критерій вартості експлуатації та формули розрахунку даного критерію за методом ТСО.

**Мета дослідження.** Основною метою є покращення процесу планування маршруту руху шляхом розробки підсистеми для формування раціонального маршруту переміщення мобільного транспортного робота при виконанні робочих транспортних операцій для умов роботи в промислових приміщеннях, складах та теплицях з урахуванням карти приміщення, технологічних позицій для виконання типових операцій.

Об’єкт дослідження: рух мобільних роботів по заданій карті з виконанням типових операцій.

Предмет дослідження: алгоритми пошуку раціонального маршруту виконання типових операцій (завантаження, розвантаження та очікування) роботом в промислових приміщеннях по заданій карті.

Для реалізації поставленої мети сформульовані наступні завдання:

- дослідження алгоритмів розбиття мапи на опорні точки для побудови графу;
- дослідження алгоритмів пошуку оптимального шляху для мобільних роботів по заданих картах;
- розробка програмного забезпечення на базі A* для переміщення мобільного робота по заданій карті з чергою відвідування технологічних позицій;
- симуляція та тестування роботи програмного забезпечення.

Наукова новизна результатів магістерської дисертації полягає в тому, що запропоновано критерії оцінки вартості володіння мобільним транспортним роботом та системою його управління, а саме критерії на базі методу ТСО. Результат досягнутий шляхом проведення порівняльного аналізу прихованих та явних витрат, які впливають на роботу мобільного транспортного робота.

Практичне значення отриманих результатів полягає в тому, що реалізовані методи та програмне забезпечення поєднані в межах однієї системи і максимально прості та суттєво дешевші у використанні для кінцевого користувача. Система є простою у розгортанні та не має специфічних вимог до технологічних приміщень, а також має високу ступінь маштабованості.

**Зв'язок з науковими програмами, планами, темами.** Робота виконувалась на кафедрі інформаційних систем та технологій Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського".

**Публікації.** Наукові положення дисертації опубліковані в:

Критерій оцінки вартості експлуатації мобільного транспортного робота / О.С. Кривич, С.А. Резніков // InfoCom 2021: Матеріали XI та XII Міжнародних науково-практичних конференцій з інформаційних систем та технологій, м. Київ, 13-14 травня 2021р., 1-2 грудня 2021р. – К.: Вид-во ТОВ "Інжиніринг", 2021. – 70с. – Мови укр., рос., англ.

Магістерська дисертація була виконана за науково-дослідною роботою № 2000/556 від 10.06.2019р. «Визначення технічних параметрів пошукових дронів для зон катастроф та стихійного лиха».

Ключові слова: РОБОТОТЕХНІКА, ROS, RVIZ, ПОШУК ШЛЯХУ, МОБІЛЬНІ ТРАНСПОРТНІ РОБОТИ.

## ABSTRACT

Explanatory note size – 102 pages, contains 44 illustrations, 24 tables, 8 applications.

**Topicality.** Examines the problem of of finding the optimal way to move a four-wheeled mobile robot in the storages with the performance of typical operations of unloading, loading and waiting. The main shortcomings of modern solutions are substantiated, namely the need for complete re-equipment of premises for the use of such systems. The need to simplify the process of deploying a traffic control system and the need to develop a methodology or criteria for estimating the cost of a search algorithm, proposed a criterion of operating costs and a formula for calculating this criterion by the TSO method.

**The aim of the study.** The main target is to improve the path planning process by developing a subsystem for the formation of a rational four-wheeled robot route when performing working transport operations for working conditions in industrial premises, warehouses and greenhouses, taking into account the room map, technological positions for typical operations.

Object of research: movement of mobile robots on a given map with the performance of typical operations.

Subject of research: algorithms for finding a rational route for performing typical operations (loading, unloading and waiting) by a robot in industrial premises on a given map.

To achieve this goal, the following tasks were formulated:

- study of algorithms for dividing the map into reference points for plotting;
- study of algorithms for finding the optimal path for mobile robots on given maps;
- development of software based on A * to move the mobile robot on a given map with a queue to visit technological positions;
- simulation and testing of software.

The scientific novelty of the results of the master's dissertation is that the criteria for estimating the cost of owning a mobile transport robot and its control system are proposed, namely the criteria based on the TSO method. The result was achieved by

conducting a comparative analysis of hidden and explicit costs that affect the operation of mobile transport robot.

The practical value of the obtained results is that the implemented methods and software are combined within one system and are as simple and significantly cheaper to use for the end user. The system is easy to deploy and has no specific requirements for technological premises, and also has a high degree of scalability.

**Relationship with working with scientific programs, plans, topics.** Work was performed at the Department of Informatics and Software Engineering of the National Technical University of Ukraine «Kyiv Polytechnic Institute. Igor Sikorsky».

**Publications.** The scientific provisions of the dissertation published in:

A criterion for estimating the cost of ownership of a mobile transport robot / Olha Kryvych, Serhii Reznikov // INFOCOM ADVANCED SOLUTIONS 2021: CONFERENCE PROCEEDINGS 11th and 12th Scientific and practical conference, KYIV, UKRAINE, May 13-14, 2021, December 1-2, 2021. – 70 pages.

The master's dissertation was performed on the basis of research work №2000 /556 dated 10.06.2019. "Determination of technical parameters of search drones for disaster zones and natural disasters."

Keywords: ROBOTICS, ROS, RVIZ, WAY SEARCH, MOBILE TRANSPORT WORKS.

ЗМІСТ

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

MP - мобільний робот

Navmesh - Navigation mesh

RT - Rapidly-Exploring Random Tree

APF - Artificial Potential Fields

VFH - Vector Field Histogram

ROS - Robot operating system

AMCL - Adaptive Monte Carlo Localization

SLAM - Simultaneous Localization and Mapping

OGM - Occupancy Grid Map

TCO - Total Cost Of Ownership

OCP - Open Close Principle

# ВСТУП

Автоматизація виробничих процесів за допомогою мобільних роботів (МР) для полегшення роботи людини вже декілька десятиліть не втрачає своєї актуальності. МР вже зайняли важливе місце в усіх сферах діяльності людини, які несуть ризики для здоров'я, включаючи перевезення вантажу на складах та інших промислових приміщеннях. На відміну від статичних промислових роботів, такий робот має рух без обмежень завдяки своєму фізичному розміру. В результаті мобільні роботи можуть працювати у великому робочому просторі, досліджувати невідоме середовище, запам'ятовувати його та зберігати мапу, і тому можуть виконувати завдання всюди, де це необхідно. Такий підхід не тільки виключає вплив людини на процеси, а й пришвидшує виконання типових операцій. Транспортні МР транспортують партії товарів на наступний етап обробки, дозволяючи працівникам швидше переходити до наступного завдання.

Планування глобальних та локальних шляхів переміщення МР є актуальною областю дослідження, оскільки це спрощує та прискорює виконання технологічних задач. Дійсно, система керування МР повинна містити в собі підсистеми пошуку шляху та побудови оптимальних маршрутів для пересування. Деякі сучасні системи керування МР включають в себе алгоритми розпізнавання перешкод та можуть в умовах реального часу пристосовуватись до змін середовища.

У робототехніці планування шляху в його найпростішій формі полягає у встановленні шляху без перешкод для робота, починаючи від початкового (або поточного) розташування до цільової точки призначення.

Розробка ефективного алгоритму планування шляхів є важливим питанням навігації МР, оскільки оптимальність шляху впливає на ефективність виконання завдання. Побудований шлях повинен відповідати набору критеріїв оптимізації, наприклад: загальну пройдену відстань роботом, яка в свою чергу може впливати на використані їм ресурси, такі як електроенергія, пальне або загальний час виконання технологічної задачі.

Метою даної роботи є покращення процесу планування шляху шляхом розробки підсистеми для формування раціонального маршруту переміщення чотириколісного робота при виконанні робочих транспортних операцій для умов роботи в промислових приміщеннях, складах та теплицях з урахуванням карти приміщення, технологічних позицій для виконання типових операцій.

**Задачі даної роботи**:

1. дослідження алгоритмів розбиття мапи на опорні точки для побудови графу;
2. дослідження алгоритмів пошуку оптимального шляху для мобільних роботів по заданих картах;
3. розробка програмного забезпечення на базі A* для переміщення мобільного робота по заданій карті з чергою відвідування технологічних позицій;
4. симуляція та тестування роботи програмного забезпечення.

**Практична частина роботи**: програмне забезпечення (плагін) з реалізованим алгоритмом переміщення мобільного робота по заданій мапі з урахуванням черги технологічних позицій.

Оформлення звітної документації здійснено відповідно до нормативних документів оформлення магістерських дисертацій кафедри Інформаційних систем та технологій Факультету інформатики та обчислювальної техніки у НТУУ “КПІ ім. Ігоря Сікорського” та ДСТУ 3008-2015.

# 1 ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ АЛГОРИТМІВ ПОШУКУ ОПТИМАЛЬНОГО ШЛЯХУ

У даному розділі розглянуто задачі для побудови шляху при заздалегідь заданій мапі. Проведено аналіз проблем навігації та роботи МР у промислових приміщеннях, таких як склади, теплиці тощо.

Під час аналізу планування шляху у промислових приміщеннях було встановлено, що мобільні транспортні роботи мають проблему з виконанням черги типових задач у середовищах, де потрібно розпізнавати технологічні позиції для завантаження вантажу, розвантаження та очікування наступної операції.

Також проведено огляд наявних алгоритмів пошуку найкоротшого шляху та методів обробки мап середовища для розв'язання проблеми прокладання оптимального маршруту. Проведено дослідження роботи та оптимальності кожного з алгоритмів в контексті розв'язання проблеми побудови шляху робота в промислових приміщеннях з технологічними позиціями. Були наведені схематичні приклади їх роботи. Обрано алгоритми за основу для модифікації та розв'язання поставленої проблеми.

## 1.1 Визначення процесу планування шляху

Навігація — це здатність мобільних роботів визначати своє положення в середовищі, де він розташований (локалізація), а також планувати та виконувати шлях до цільового розташування. МР можуть бути використані в різних середовищах для широкого спектра застосувань і виконувати навігацію без впливу на те, чи будуть вони використовуватися вдома чи на сільськогосподарських полях. Незалежно від сфери застосування роботу необхідно переміщуватися в середовищі свого функціонування. Для цього використовується система навігації. Навігація мобільного робота охоплює великий діапазон різних технологій та застосувань.

Систему навігації в робототехніці поділяють на три рівні:

- глобальна - визначення абсолютних координат пристрою при русі довгими маршрутами;

- локальна - визначення координат пристрою по відношенню до деякої (зазвичай початкової) точки. Ця схема затребувана розробниками тактичних безпілотних літаків та наземних роботів, що виконують місії в межах наперед відомої області;
- персональна - позиціонування роботом частин свого тіла та взаємодія з прилеглими предметами, що актуально для пристроїв, забезпечених маніпуляторами.

Завдяки значній частці досліджень у галузі навігації, її можна поділити на чотири основні категорії:

- позиціювання;
- планування шляху (маршрутизація),
- планування руху (контроль);
- створення мапи.

З метою ведення навігації, робот повинен займати позиції, опис яких може бути як кількісним (координати з точки зору одиниці довжини), так і якісним (наприклад, кольорові візерунки тощо). Після цього робот повинен визначити шлях, що залежить одночасно від його поточного положення, аналізування даних з навколишнього середовища (з мапою чи без неї), а також від кінцевої цілі. Отже, планування шляху — це завдання оцінки шляху в середовищі, яке дозволяє досягти бажаної мети [1]. Лише після визначення шляху роботом слід керувати таким чином, щоб він міг рухатися за цим шляхом. Управління рухом робота веде до методології планування руху. Деякі роботи створюють мапу свого навколишнього середовища, щоб їм було зручно використовувати її для наступних цілей маршрутизації. Посилаючись на інформацію про місцезнаходження МР, надану з його середовища, проблеми маршрутизації поділяються на три великі категорії:

- маршрутизація у відомому середовищі;
- маршрутизація в частково відомому середовищі;
- маршрутизація в невідомому середовищі.

Щодо випадку відомого середовища, потрібно використовувати весь спектр зору, який забезпечує камери або лазери, тоді роботу надається повна інформація

про середовище у вигляді файлу мапи та велика кількість інформації, що стосується побудови маршруту [1].

У частково відомому середовищі робот мапу навколишнього середовища на першому кроці. Наступна інформація про перешкоди, які з’являються у полі зору робота (включаючи людей або тварин) або будь-які зміни в приміщенні, яких немає на мапі, потребують оновлення мапи для побудови оптимального маршруту робота.

У невідомому середовищі робот не має карти і замість цього використовує інформацію, отриману від датчиків.

Складання детальної мапи з урахуванням всіх шляхів та перешкод на маленькому та середньому за розміром приміщенні, не є складною задачею. Проте зі зростанням площі та кількості перешкод, ресурси, які потрібні для побудови мапи, різко зростають та ускладнюють її створення [4].

## 1.2 Визначення проблеми побудови маршруту в умовах промислових приміщень

Планування шляху є критично важливим питанням у сфері робототехніки та, загалом, у сфері автоматизації. У рамках даної роботи розглянута проблема побудови шляху мобільного транспортного робота для роботи у промислових приміщеннях (склади, теплиці тощо) з умовою почергового відвідування технологічних позицій для виконання типових операцій завантаження вантажу, розвантаження та очікування наступної операції.

Сучасні роботи при досягнення деякої поставленої цілі, або точки в просторі, опираються на свою поточну позицію, позицію цілі та показники з різних датчиків, таких як дальноміри та інше, та слідують у напрямку до заданої точки, минаючи перешкоди. На жаль, такий підхід не завжди є оптимальним, або навіть можливим, якщо точка знаходиться за перешкодою, яку неможливо об'їхати, що в свою чергу може викликати повну зупинку технологічного процесу тощо, якщо мова йде про промислові приміщення [8].

При  виконанні черги послідовних операцій на визначених технологічних позиціях по відомій мапі, постає проблема створення мапи середовища, визначення усіх ключових позицій, подальше їх маркування та запам’ятовування системою керування роботом, а також побудова оптимального шляху залежно від поточної задачі.

В сучасних умовах для побудови мапи промислового приміщення транспортному роботу навряд чи потрібно об'їжджати та сканувати його для отримання мапи (рис 1.1), адже це може бути небезпечно для самого робота, вантажів, які розташовані в приміщенні, обладнання та персоналу. Тому для таких цілей можна використати малогабаритного робота або дрона для отримання первинної карти. Після цього потрібно завантажити її в систему управління роботом та скласти граф доріг з урахуванням технологічних позицій, розмірів транспортних роботів та інших параметрів середовища. Таким чином, робот отримує інформацію про кожну технологічну ділянку, а також оптимальний маршрут руху без небезпеки для самого обладнання, вантажу та персоналу.

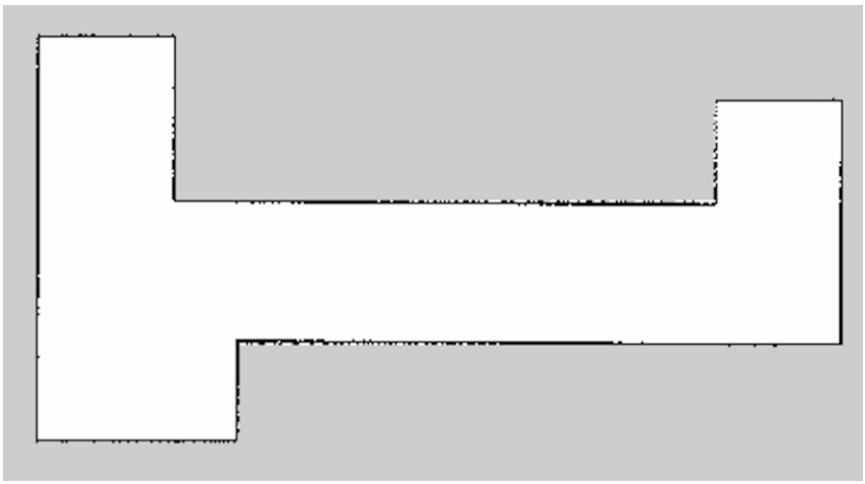

Рисунок 1.1 — Приклад мапи приміщення

Важливим критерієм розв'язання поставленої задачі пошуку оптимального маршруту складає час виконання циклу задач. Висока швидкість роботи для зменшення часу може перешкоджати точності та повторюваності руху та бути небезпечною для вантажу, який перевозить робот, оскільки від виконавчих механізмів та системи управління потрібні надзвичайні характеристики. Тому слід

приділити увагу створенню траєкторії, яка могла б бути виконана за відносно малий час, але водночас безпечна для робота, з точки зору уникнення надмірних прискорень приводів та вібрацій механічної конструкції.

## 1.3 Алгоритми побудови графу шляхів

Навігація і позиціювання всередині приміщення являє собою окрему, досить складну, задачу. Власне створення мапи – це процес отримання інформації про навколишнє середовище за допомогою сенсорних даних і представлення інформації у форматі, зрозумілому для робота. Отримана карта середовища може бути використана роботом для покращення його ефективності в навігації. Позиціювання здійснюється для деякої мапи (схеми приміщення), і може бути прив'язане як до географічних координат, так і до умовних локальних координат мапи [13].

Щоб спланувати шлях, потрібно якимось чином представити середовище в комп’ютері. Розрізняють два підходи: дискретне та безперервне наближення. У дискретному наближенні мапа поділяється на частини рівних (наприклад, сітка) або різних розмірів (наприклад, кімнати в будівлі). Останні мапи також відомі як топологічні карти. Дискретні мапи добре підходять для представлення в графі. Тут кожні фрагменти відповідають вершинам, які з’єднані ребрами, тому робот може переміщатися від однієї вершини до іншої. Наприклад, дорожня карта — це топологічна карта з перехрестями як вершинами і дорогами як ребрами. З точки зору обчислень, граф може зберігатися у вигляді списку або матриці суміжності/інцидентності. Безперервна апроксимація вимагає визначення внутрішніх перешкод і зовнішніх границь, як правило, у формі багатокутників, тоді як шляхи можуть бути закодовані як послідовності дійсних чисел.

Планування шляху займає основні позиції в робототехніці та навігації. Здебільшого весь цей процес можна розділити на два окремі кроки. Першим кроком потрібно здійснити дискретизацію безперервного середовища, представити його в зрозумілій структурі (як правило, у вигляді графа) та виконати пошук шляху з

заданого місця на карті. Другим кроком, планування шляху на графі маршрутних точок можна розглядати як задачу пошуку шляху на графі [13].

На поточний день існує декілька алгоритмів розбиття мап на графи для подальшого їх використання в плануваннях маршрутів. Розбиття методом сітки та методом поля потенціалів та інші часто використовуються для побудови графів переміщення. Ці підходи зазвичай вимагають надзвичайно точної інформації від датчиків для отримання початкового графа, а також можуть мати значний час розрахунку.

### 1.3.1 Граф видимості

У обчислювальній геометрії та плануванні руху роботи граф видимості — це граф невидимих місць, як правило, для набору точок і перешкод в евклідовій площині [14]. Кожен вузол на графіку представляє розташування точки, а кожне ребро — видимий зв'язок між ними. Тобто, якщо відрізок лінії, що з'єднує два вузла, не проходить через жодну перешкоду, між ними на графі проводиться ребро (Рис. 1.2).

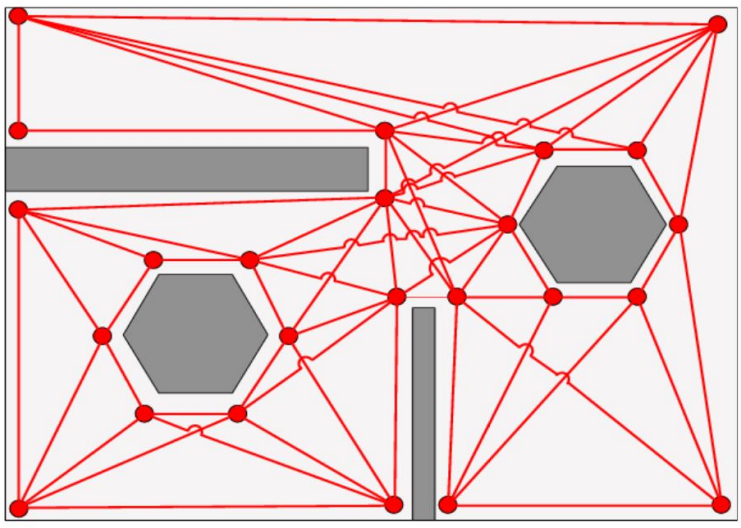

Рисунок 1.2 — Приклад графу видимості

Граф видимості можна використовувати для пошуку найкоротших шляхів серед багатокутних перешкод на площині: найкоротший шлях між двома перешкодами відповідає відрізкам прямої лінії. Отже, евклідова задача про

найкоротший шлях може бути розбита на дві прості підзадачі: побудова графу видимості та застосування до графа алгоритму найкоротшого шляху, такого як алгоритм A*, Дейкстри та інших. Для планування руху робота, який має невеликий розмір у порівнянні з перешкодами, подібний підхід може бути використаний після абстрактного розширення перешкод (збільшення їх віртуального розміру), щоб компенсувати розміри робота.

### 1.3.2 Граф маршрутних точок

Щоб керувати процесом планування маршруту, потрібно мати певні знання та математичне представлення навколишнього середовища. У зв'язку з цим середовище, яке зазвичай представляється у вигляді полігонів або зображення непрохідних об’єктів тощо, важко використовувати через його величезні розміри та відсутність зв’язку між цими полігонами, об'єктами та іншим. Це спонукає дослідників та інженерів використовувати прості структури даних, які часто називають мапами.

Одним із загальновживаних зображень мапи є граф маршрутних точок (рис. 1.3), який складається з набору точок маршруту. Вони представляють важливі точки у деякому середовищі, такі як кути та ребра, що з'єднують ці точки маршруту. Ця структура даних проста, але вона часто створюється вручну, і тому потребує автоматизації для використання в умовах конкретної задачі [15].

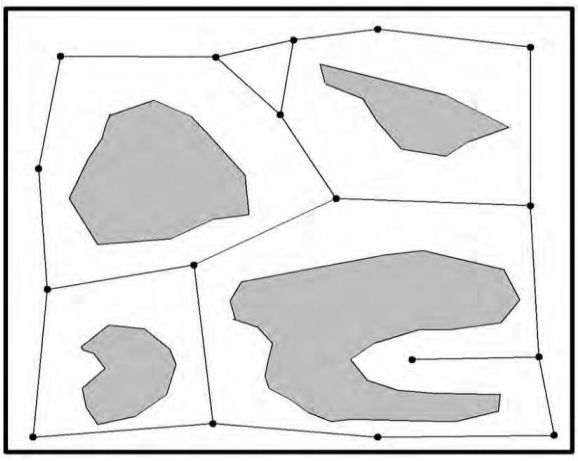

Рисунок 1.3 — Приклад графу маршрутних точок

### 1.3.3 Методи розбиття мапи на сітку

У підходах сітки вільний простір поділяється на набір простих комірок, і розраховуються відношення суміжності між комірками. Осередки можуть бути об'єктно-залежними або об'єктно-незалежними. У об'єктно-залежному методі розкладання межі перешкод використовуються для генерування меж комірок, а об'єднання вільних комірок безпосередньо вільного простору [16].

Кількість комірок, що представляють простір, невелика, але складність розкладання велика, а обчислення наявності об’єктів в них, перетину, зв’язку та суміжності дуже складні. Для порівняння, об'єктно-незалежні методи розкладання, такі як метод сітки та метод дерева квадрантів, ефективні у практичному застосуванні. Однак, оскільки форма та місце розташування комірки визначаються незалежно від форми та розміщення об’єктів-перешкод, комірки не можуть чітко зображати межі перешкод. Це призводить до того, що багато маленьких клітин знаходяться біля меж перешкод (рис 1.4).

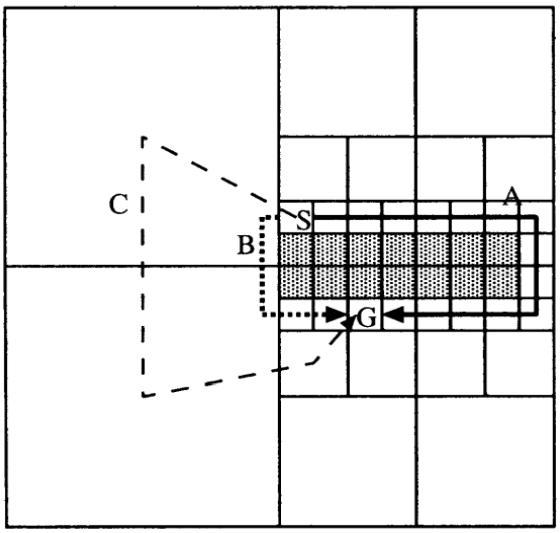


Рисунок 1.4 — Приклад побудови дерева квадрантів

Метод сітки гарантує знаходження найкоротшого шляху з точністю до роздільної здатності сітки. Однак метод сітки має багато комірок, тому етап планування шляху не є ефективним. Дерево квадрантів створюється шляхом рекурсивного розкладання двовимірного (2D) середовища на області однакового розміру. Це зменшує кількість дрібних комірок у вільному просторі.

Кількість комірок для пошуку в дереві квадрантів можна зменшити за допомогою подання з різною роздільною здатністю. Клітинку у вільному просторі називають «білою», а клітинку у просторі перешкод - «чорною» (Рис. 1.5). Ділянка, що містить суміш білих і чорних комірок, називається “сірою” коміркою.

Якщо сіру клітинку обрано як частину оптимального шляху, вона перетворюється на чорну або білу клітинку. Використовуючи це адаптивне переміщення по дереву квадрантів, можна прискорити планування шляху. Однак схема повного вирішення не може скоротити час планування шляху, якщо робот рухається по вузьких проходах біля перешкод. Іншим рішенням є використання об’єктно-залежного розкладання сітки. Однак для цього потрібна вхідна сітка з набагато вищою роздільною здатністю, ніж дерево квадрантів, та значно більшим часом обчислень, оскільки об'єктно-залежний алгоритм розкладання комірок передбачає складні обчислення.

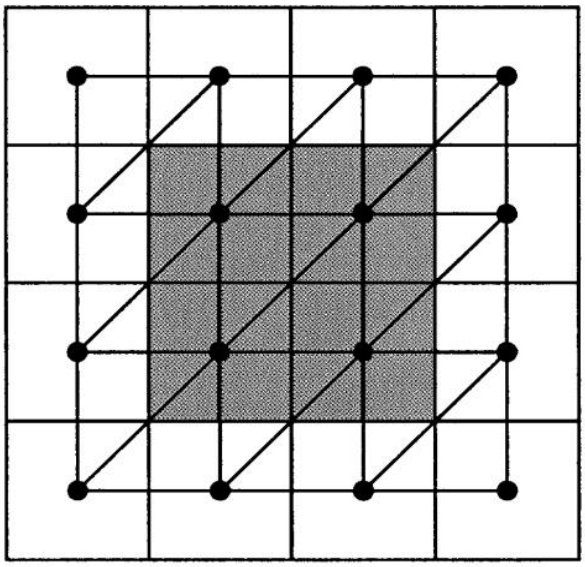

Рисунок 1.5 — Розбиття мапи на чорні та білі комірки за методом сітки

## 1.4 Огляд алгоритмів пошуку найкоротшого шляху

Планувальник маршрутів зазвичай має під собою алгоритм планування шляху або пошуку шляху, який стає важливим компонентом у робототехніці та автономних транспортних засобах. Хоча завдання пошуку шляху з точки зору середовища та методів пошуку в основному ідентичні, параметри та обмеження,

що накладаються на пошук шляху, значно відрізняються в різних областях. Наприклад, навіть обидва терміни пошуку шляху та планування шляху взаємозамінно використовуються в різних областях. Хоча в багатьох програмах проблема обмежується пошуком найкоротшого можливого шляху між точками A і B, зазвичай це не так у контексті пошуку шляхів на бездоріжжі, де немає доріг чи точок шляху.

Протягом десятиліть було запропоновано багато алгоритмів планування шляхів, таких як алгоритм Дейкстри, алгоритм A*, агресивний евристичний пошук (AHS), пошук точки переходу (JPS), алгоритм D*, Lifelong Planning A*, D* Lite, Adaptive A*, Path-Adaptive A*, Tree Adaptive A*, Navmesh, Q-навчання тощо. Серед цих алгоритмів алгоритм A*, AHS і JPS можна використовувати лише для планування маршруту з однією позицією, алгоритм D*, LPA*, D* Lite, AA*, Path-Adaptive A*, Tree Adaptive A* підходять для планування шляху в частково відомих середовищах, алгоритм Дейкстри, Q-Learning можуть бути використані для планування шляху до довільно вибраної початкової позиції до єдиної цільової позиції. Q-Learning потрібно обчислювати ітеративно, щоб переконатися, що всі значення стану зібрані [2].

Детально виконано розгляд та визначення переваг і недоліків декількох наступних алгоритмів та методів пошуку шляху:

1. A*
2. Navmesh (Navigation mesh)
3. Rapidly-Exploring Random Tree (RRT)
4. Probabilistic Roadmap (PRM)
5. Artificial Potential Fields (APF)
6. Vector Field Histogram (VFH)

### 1.4.1 Алгоритм A*

A* - це евристичний алгоритм пошуку, заснований на алгоритмі Дейкстри. Евристичний пошук — це пошук, призначений для того, щоб оцінити кожен вузол розширення у просторі станів, вибрати найкраще місце розташування вузла, а потім здійснювати пошук із початкового вузла, поки він не знайде цільовий вузол. У евристичному пошуку місце оцінки є дуже важливим, і використання різних оцінок доцільності того чи іншого шляху може мати різні ефекти [3].

Функція оцінки представляє собою вартість переходу від поточного вузла в цільовий вузол, що є евристичним. У проблемі пошуку шляхів зазвичай використовується функція оцінки Манхеттена для оцінки вартості. Функція оцінки поточного вузла $n$ виражається як (1):

$$F(n) = H(n) + G(n) \tag{1}$$

де $F(n)$ - функція оцінки;

$H(n)$ - евристичне значення найкоротшого шляху будь-якого вузла $n$ до цільової точки;

$G(n)$ - найкоротший шлях від початкової точки до будь-якого вузла $n$.

Можна довести, що якщо функція оцінки задовольняє умові сумісності, що функція $H(n)$ менша за фактичну вартість вузла $n$ до цільового вузла, то алгоритм A* може знайти оптимальний шлях.

Перед використанням алгоритму A* карту слід розділити на фрагменти. Чотиригранний поділ є найпростішою формою, оскільки деякі ділянки на мапі можуть бути розділені на опуклу карту багатокутників тощо.

Далі потрібно знайти найкоротший шлях від A до B, а чорна частина є перешкодою посередині карти (рис. 1.6).

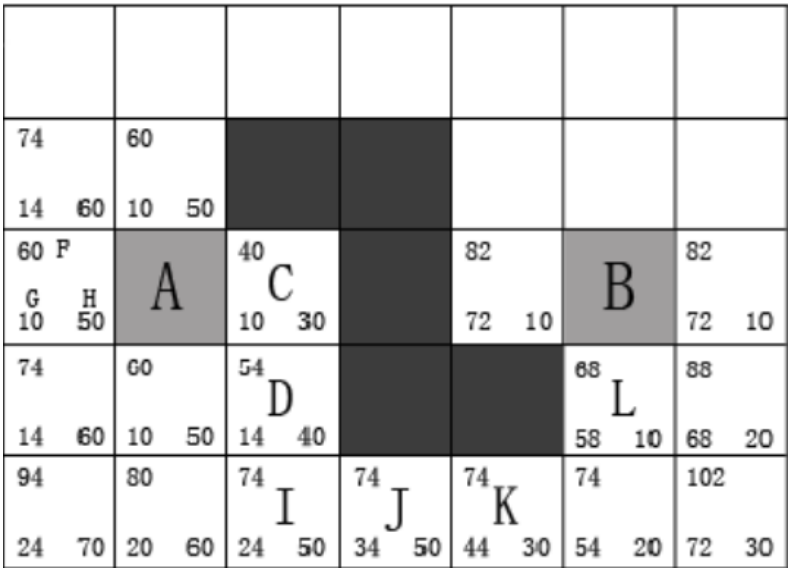


Рисунок 1.6 — Результат пошуку шляху від A до B за алгоритмом A*

Відстань між діагональними вузлами дорівнює 14, а відстань між правими кутами — 10. Значення F вказано у верхньому лівому куті, G позначено ліворуч знизу, а H - праворуч. Під час пошуку алгоритм оцінює та розраховує вартість переходу від кожного вузла в інший у напрямку цільової точки, обираючи найкращу з них. Ці дії повторюються, доки не буде досягнуто цілі, або доки не будуть відвідані всі доступні вузли.

Після пошуку та знаходження найкоротшого шляху, рухаючись назад по оптимальному за оцінками маршруту можна побудувати маршрут переміщення від A до B, та отримати шлях A, D, I, J, K, L, B.

Алгоритм A* у теорії має найкоротший час роботи, але також має свої недоліки. Його просторове зростання є експоненціальним, тому потрібна подальша оптимізація. Це може змусити поступово збільшувати дані таблиці, знаходячи мінімальне значення у всіх вузлах шляхом постійного пошуку колекції, що призводить до того, що процес пошуку стає все повільнішим. Щодо проблеми зчитування даних, деякі вчені запропонували алгоритм A* на основі зони обмеженого доступу для зменшення завантаження даних.

### 1.4.2 Алгоритм Navmesh

Так само як з алгоритмом A*, для Navmesh також потрібно розділити мапу на багатокутник з вузлами. Алгоритм A* буде добре працювати, якщо багатокутник правильний, але для Navmesh це не обов’язково. Light irradiation - це найпоширеніший алгоритм, який використовується при пошуку шляхів за допомогою Navmesh. Як показано на рис. 1.7, навігаційна сітка складається з довільних опуклих багатокутників [5].

Сіра сітка являє собою недоступну область, біла сітка представляє область, в яку можна потрапити. Усі вершини сітки зберігаються в порядку годинникової стрілки.

Як показано на рис. 1.7, потрібно обчислити опуклий багатокутник, де P0 відвідано, якщо P0 досягає T0 у ділянці D. По-перше, необхідно розрахувати відношення між P0 і багатокутниками, перетин або паралель, і напрямок променя, якщо вони перетинаються.

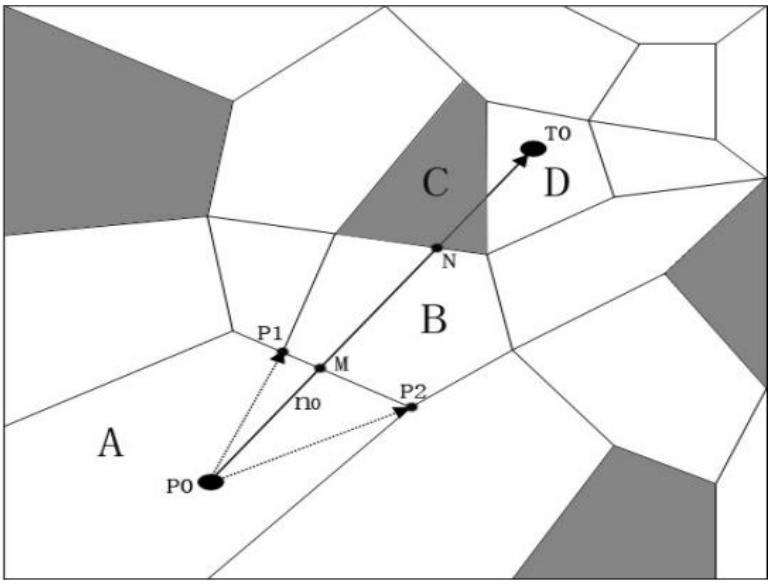


Рисунок 1.7 — Приклад пошуку шляху за Navmesh

Взявши за приклад ділянку A і промінь r =(p0,n0), визначається зв'язок між A і променем r. Обходячи A з кожної сторони багатокутника, отримано точку початку і кінця кожної сторони. Початкове значення вектору можна розрахувати за формулою (2):

$$start = \overline{(P0,P1)} \rightarrow\times n0 \quad (2)$$

де $start$ – початкове значення вектору;

$P0$ – початкова точка;

$P1$ – кінцева точка;

$n0$ – кінцеве значення променю r.

Встановити кінцеве значення вектору, як показано у формулі (3):

$$end = \overline{(P0,P2)} \rightarrow\times n0 \quad (3)$$

де $end$ – кінцеве значення вектору;

$P0$ – початкова точка;

$P2$ – кінцева точка;

$n0$ – кінцеве значення променю r.

Далі порівнюються напрямки двох векторів. Якщо вони однакові, вектор початку та кінця знаходяться на одній стороні n0, якщо ні, ребро не перетинається з променем r. Якщо start > 0 і end < 0 істинне, тоді це ребро вихідна сторона багатокутника A, перетинається з променем r. Якщо start < 0 і end > 0 істинне, то ребро є перетином багатокутника A, який перетинається з протилежним напрямком променя r. Визначивши вихідну точку та ребро у проміні r, що перетинається з багатокутником, визначається, чи інше бічне ребро багатокутника є вхідним. Якщо виявляється, що ця ділянка є перешкодою, розрахунок пропускається та береться інша сторона ділянки. Цей процес повторюється послідовно, поки не знайдеться цільова точка [6].

### 1.4.3 Алгоритм RRT

RRT - це імовірнісний алгоритм. Алгоритм, як правило, дає ефективні результати у невипуклих ділянках. Це поширений метод планування шляху роботів, його початкова ідея алгоритму дуже проста: якщо вихідна точка є початковим вузлом, випадкове розширення дерева створюється шляхом

випадкового вибору та додавання кінцевих вузлів. Коли кінцевий вузол в випадковому дереві містить цільову точку або входить в цільову область, його можна знайти в випадковому дереві. Дерево будується поступово, і дві різні функції, які є створенням і розширенням дерева, використовуються для всієї мапи. Розширення дерева одностороннє — від початку до кінця (рис. 1.8). Існує і розширення дерева двонаправленим способом, який покращує ефективність часу всього рішення. Цей метод називається двонаправленим RRT (bRRT), і дерево встановлюється як в початковій, так і в цільовій точках, а потім розширюється на невидимі частини. Процес пошуку припиняється, коли знайдена точка стику цих двох дерев і цей шлях, який є виходом алгоритму, є оптимальним маршрутом [7].

Рисунок 1.8 — Приклад побудови дерева за алгоритмом RRT

Відповідно до інших алгоритмів планування шляху, RRT виробляє просторове моделювання, виконує виявлення зіткнень у точці вибору, і може ефективно вирішувати проблеми планування шляху в багатовимірному просторі та складних обмеженнях. Цей метод може швидко й ефективно виконувати пошук у просторі великих розмірів. Через випадкові точки вибірки в просторі станів пошук направляється в пусту область, щоб знайти запланований шлях від початкової точки до цільової точки. Алгоритм підходить для рішень роботів з декількома ступенями свободи в складних приміщеннях.

### 1.4.4 Алгоритм PRM

Одним з імовірнісних алгоритмів планування маршруту, є імовірнісна дорожня карта (PRM). PRM - це метод пошуку на основі графа, який перетворює безперервний простір у дискретний, а потім використовує інші алгоритми пошуку, щоб знайти шлях на маршруті для підвищення ефективності пошуку (рис. 1.9).

PRM, заснований на технології випадкового вибору, може ефективно вирішити проблему планування шляху в багатомірному просторі при складних обмеженнях. Оптимальний шлях визначається шляхом обчислення відстані ребер, які є зв’язками між випадково створеними вузлами на карті. Перш за все, вузли генеруються шляхом вибірки з точок без перешкод на карті, а потім ці вузли зв’язуються між собою, і, нарешті, оцінюється вартість шляху. Швидка стратегія випадкових кластерів може забезпечити швидші результати, ніж інші алгоритми, але не гарантує найкоротший шлях [9].

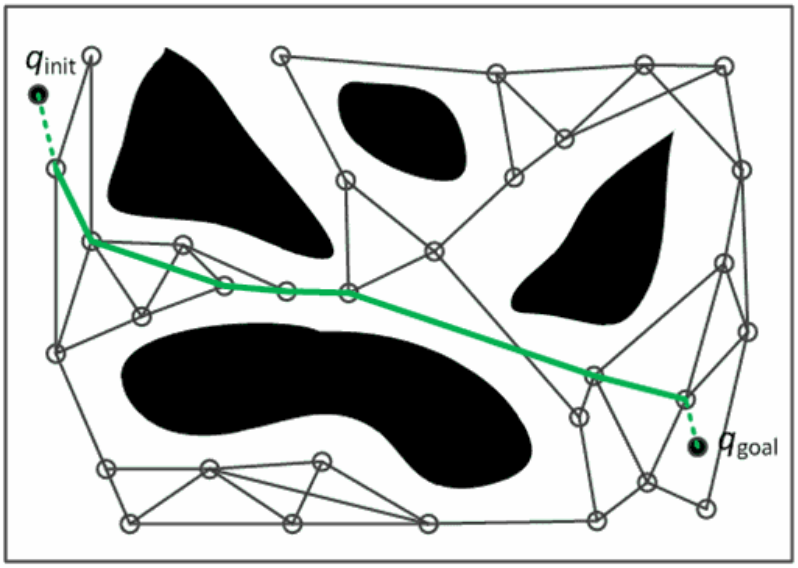


Рисунок 1.9 — Приклад пошуку шляху за алгоритмом PRM

### 1.4.5 Алгоритм APF

Метод потенційного поля (Artificial Potential field) – це один із алгоритмів у методі планування шляху робота, заснований на принципі потенціальної сили притягання або відштовхування, при якій робот та перешкода діють як позитивний заряд, а мета (точка призначення) — як негативний (рис. 1.10). Таким чином,

перешкоди відштовхуються від робота, створюючи силу відштовхування, а кінцева точка притягує робота через протилежний заряд, що призводить до сили притягання. Кінцева сила — це векторна сума всієї сили відштовхування та притягання [10].

Для статичних перешкод у відомому середовищі потенціал оцінюється в автономному режимі, тому швидкість роботи алгоритму в межах енергетичного поля від початку до точки призначення не змінюється.

Недоліком є те, що метод потенційних полів страждає від проблеми локальних мінімумів та високої складності через свою двоопераційну модель шляху.

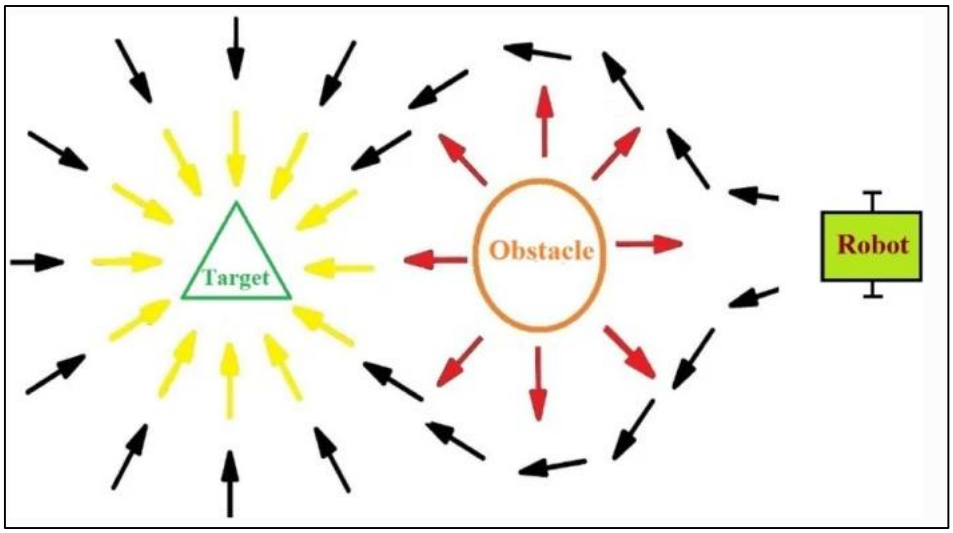


Рисунок 1.10 — Процес пошуку шляху за методом APF

Визначено деякі проблеми, які притаманні потенційним полям, які існують у всіх реалізаціях методу. Серед таких проблем можна також виділити відсутність місця між перешкодами та коливання робота при наближенні до перешкоди [11].

Проблема з локальними мінімумами може виникнути, коли все векторне поле від перешкод та точки призначення перекривають одне одного, щоб робот ніколи не досяг мети. Проблему локальних мінімумів можна подолати, повернувшись із вихідної позиції або зробивши деякі випадкові рухи з поточного розташування або використовуючи будь-який з Bug алгоритмів для уникнення перешкод. Ця проблема локальних мінімумів також може бути вирішена за допомогою деяких

методів м'яких обчислень[1]. Були проведені роботи над плануванням шляху для мобільного робота за допомогою APF, в якому робоча область для мобільного робота поділена на прямокутні комірки сітки, де кожна клітина розглядається як перешкода чи вільна ділянка. Було оцінено потенційну функцію для кожної комірки на основі відстані, яку автономний мобільний робот проїхав від джерела, перешкод та місця призначення.

Наступною проблемою є відсутність проходу між близько розташованими перешкодами. Якщо дві перешкоди розміщені близько одна до одної, наприклад як дверний отвір, сили відштовхування від кожної перешкоди об’єднуються в єдину силу відштовхування, яка вказує напрям подалі від отвору між перешкодами. Після цього робот відвертається від проходу, навіть якщо мета знаходиться з іншого боку перешкоди.

Варто згадати проблему коливання за наявності перешкод та у вузьких проходах. Було досліджено, що робот іноді буде коливатися за наявності перешкод у вузьких коридорах. Коли мобільний робот рухається вздовж центральної лінії між двома стінами, рух є стабільним. Однак робот буде дрейфувати ближче до однієї зі стін, оскільки він буде відштовхнутий назад через центральну лінію силою відштовхування від найближчої стіни. Потім робот буде відштовхнутий від іншої стіни та відкинутий через центральну лінію. Така поведінка продовжиться, і робот буде коливатися між двома стінами. Ці проблеми можна подолати методами м’яких обчислень.

### 1.4.6 Алгоритм VFH

VFH був запропонований як алгоритм, який розглядає недоліки потенційних полів. Цей метод використовує сітку гістограми для представлення середовища, в якому рухається робот (рис. 1.11). Вміст кожної клітинки в сітці гістограми розглядається як вектор перешкоди. Напрямком вектору є напрямок від поточного

[1] М'які обчислення - термін, що позначає сукупність неточних, наближених методів розв'язування задач, які часто не мають рішення за поліноміальний час. Технологія м'яких обчислень здатна вирішувати завдання управління слабо структурованими об'єктами управління.

положення робота до перешкоди. Після цього територія, що оточує робота, ділиться на ряд рівних секторів, і в кожному секторі вектори перешкод підсумовуються разом, щоб надати щільність полярних перешкод [12].

Проблеми методу VFH пов’язані з динамічними обмеженнями. Динамічні обмеження, накладені на робота, блокують деякі сектори, які були позначені як вільні для подорожей.

Налаштування параметрів, ще одна серйозна проблема методу VFH, полягає в тому, що межі, які визначаються, коли сектор зберігається для руху, визначає поведінку робота. Занадто низькі межі призводять до того, що робот не може рухатися за наявності перешкод, тоді як занадто великі межі змушують робота зіткнутися з перешкодами.

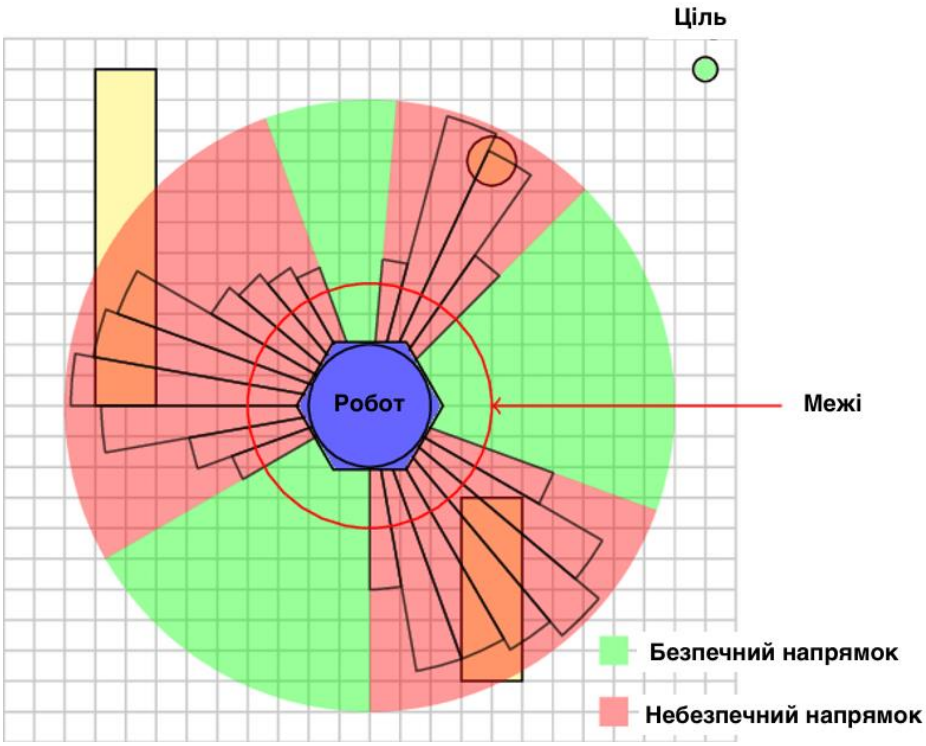


Рисунок 1.11 — Розпізнавання безпечних та небезпечних напрямків руху за методом VFH

Також був запропонований вдосконалений метод VFH для планування шляху роботи в реальному часі, коли середовище невідоме та динамічне. Параметр співвідношення для площі, яку потрібно охопити, запропоновано з урахуванням розміру робота та перешкод. Також використовувався модуль нечіткої логіки для

ефективного уникнення перешкод у динамічному середовищі. Запропонована методика виправляє накопичену помилку, потребу у великому просторі сітки для зберігання, а також долає проблему локальних мінімумів [12].

## 1.5 Постановка задачі

У результаті проведеного дослідження планування оптимальних шляхів у промислових приміщеннях, проаналізовано відомі алгоритми розбиття мапи та пошуку найкоротшого шляху. Визначено, що ці алгоритми не можуть забезпечити вирішення проблеми побудови шляху з додатковими умовами виконання операції та вимагають модифікації.

В ході аналізу опису функціонування алгоритмів побудови найкоротших шляхів та алгоритмів для створення графів переміщення на карті, було обрано алгоритм A*, за його відносну простоту, швидкодію та можливість у разі необхідності, легко його модифікувати. Для побудови графу маршрутів руху було обрано алгоритм розбиття карти на сітку, з урахуванням того, що середовище, в якому буде знаходитись робот, складається лише з об'єктів відносно правильної форми. Це є оптимальним для такого класу алгоритмів, та усуває потребу в складніших методах, таких як карта видимості або граф маршрутних точок. Обґрунтування вибору модифікації для алгоритму A* будуть описані у Розділі 4.

Метою даної роботи є створення програмного забезпечення для управління чотириколісним роботом з реалізованим алгоритмом пошуку оптимального маршруту руху для умов роботи в промислових приміщеннях з урахуванням мапи приміщення та технологічних позицій.

Для досягнення мети роботи потрібно розв'язати наступні задачі:

1. аналіз алгоритму A* (планування шляху) та методу сітки (створення графу доріг);
2. формування черги технологічних позицій для виконання типових операцій;
3. розробка програмного забезпечення (плагіну);

4. симуляція процесу пошуку оптимального шляху.

## 1.6 Висновки до розділу

У літературі на проблему планування шляху впливають два фактори: середовище, яке може бути статичним або динамічним, знання або інформація, які робот має про навколишнє середовище. Якщо робот має повне знання про навколишнє середовище, ця проблема відома як планування глобального шляху. У цій роботі розглядається проблема планування глобальних шляхів у статичних середовищах, таких як промислові приміщення.

Було наведено опис деяких найбільш популярних та придатних до використання в контексті поставленої задачі алгоритмів розбиття довільних мап на графи шляхів, та алгоритми пошуку шляхів по даний графам доріг.

Загалом у даному розділі було розглянуто [1-16] джерел. У роботах [1-3,8] проведено загальний огляд процесу планування шляху. У роботах [4-7,9,12] проаналізовано огляд та аналіз алгоритмів пошуку оптимального шляху. У роботах [10-11, 13-16] розглянуто існуючі методи розбиття мап для побудови графу шляху. У результаті було обрано алгоритм розбиття карти методом сітки та алгоритм A* для пошуку шляху, що обґрунтовано у пункті 1.5.

# 2 РОЗРОБКА АЛГОРИТМУ РУХУ РОБОТА В ПРОМИСЛОВИХ ПРИМІЩЕННЯХ

У даному розділі розглянуто проектування архітектури та розробку алгоритму розв'язання проблеми побудови оптимального маршруту руху робота в умовах промислових приміщень, з почерговим відвідуванням технологічних позицій для виконання типових операцій (завантаження, розвантаження та очікування наступного завдання). У цьому розділі будуть описані та виконані з першої по третю задачі, поставлені у пункті 1.5.

Приміщення, по якому буде пересуватися робот, є наперед відомим та складається з правильних прямокутних форм. Об’єкти в ньому також мають правильну прямокутну форму. Рух робота здійснюється по наперед згенерованому графі шляхів, який отримано обробивши мапу приміщення. Задачі, які повинен виконувати робот, подаються йому у вигляді списку точок для відвідування, а це означає, що можна інтерпретувати дану задачу як пошук найкоротшого шляху від деякої поточної позиції робота до наступної точки й так далі, поки точки, які потрібно відвідати не закінчаться, після чого відбувається повернення у вихідну точку.

## 2.1 Проектування архітектури плагіну

Перш за все потрібно визначитись з архітектурними особливостями плагіну та його взаємодії з іншими компонентами системи. В якості архітектурного рішення запропоновано використовування програмних модулів, які будуть розроблені окремо, але будуть тісно пов’язаними в контексті обміну інформацією між собою, а саме – поточними технологічними позиціями, позицією робота, графом доріг та мапою. Це, в свою чергу, полегшить тестування та відладку програмних частин та дасть змогу в подальшому використовувати дані програмні модулі в інших проектах або в інших цілях.

Процес розробки плагіну можна розбити на такі етапи:

1. Розробка архітектури програмного застосунку (плагіну): розробка архітектури має під собою мету проектування модулів системи, їх взаємодії між собою та задач для забезпечення простоти масштабування такої системи в майбутньому, розробки її окремих компонентів без жорсткої прив’язки один до одного, відладки та тестування;
2. Написання “малих” виконавчих модулів: написання виконавчих модулів системи означає безпосередню розробку програмних одиниць системи, які будуть виконувати поставлені їм задачі згідно з розробленою архітектурою. До “малих” виконавчих модулів відносяться: інструмент розміщення технологічних позицій та модуль керування рухом робота;
3. Написання основного модуля розрахунку маршруту: основний модуль представляє собою програмну одиницю, задачею якої є збір даних з різних топіків системи та інструменту розміщення технологічних позицій, вивід цих даних та надання інтерфейсу оператора для керування системою в цілому, а також передачу виконавчих команд на модуль керування рухом работа;
4. Тестування модулів: тестування модулів являє собою їх відлагодження та перевірку окремо від інших для виявлення можливих проблем та несправностей перед їх безпосередньою інтеграцією в загальну систему;
5. Тестування системи: загальне тестування системи проводиться після повної перевірки справності її окремих компонентів та являє собою набір тестів для виявлення несправностей під час взаємодії компонентів системи, їх можливі конфлікти та неправильну поведінку під час передачі інформації, керуючих команд тощо.

Для проектування хорошої архітектури програмного додатку, слід керуватися принципами розробки архітектури, а саме відкритті-закриття, заміщення Ліскоу, єдиної відповідальності, відділення інтерфейсу, інверсії залежностей.

### 2.1.1 Принцип відкриття-закриття (OCP)

Програмні компоненти такі як класи, модулі та функції повинні бути відкриті для розширення, але закриті для змін. Це можете бути обговорено, коли йде процес розробки та проєктування класів, щоб впевнитися, що коли буде потрібно розширити поведінку, не знадобиться змінювати клас, а тільки доповнювати його функціонал. Подібний принцип застосовується для модулів, пакетів і бібліотек. Якщо є бібліотека, що складається з множини класів, то є багато причин того, чим розширення функціоналу є кращим за зміну коду, який вже написаний (заради зворотної сумісності, повернення до попереднього тестування і т.д.). Це причина, по якій розробники  повинні бути впевнені, що модулі системи слідують Принципу відкриття-закриття. Стосовно класів цей принцип може бути реалізований шляхом використання абстрактних класів і конкретних класів для реалізації їх поведінки. Ця причина буде змушувати реалізовувати конкретні класи, що розширюють абстрактні класи замість зміни функціоналу. Деякі приватні випадки цього принципу являють собою патерн Шаблон і патерн Стратегія.

### 2.1.2 Принцип заміщення Ліскоу

Похідні класи повинні бути здатні повністю заміщатися їх базовими класами. Цей принцип є всього лише розширенням Принципу відкриття-закриття, що означає, що розробники повинні бути впевнені, що нові похідні класи є розширенням базових класів без зміни їх поведінки.

### 2.1.3 Принцип єдиної відповідальності (Single Responsibility Principle)

Клас повинен мати тільки одну причину для зміни. У цьому контексті відповідальність розглядається як єдина причина для зміни. Цей принцип стверджує, що якщо існує дві причини для зміни, то розробники повинні розділити функціональність на два класи. Кожен клас повинен мати тільки одну відповідальність, і в майбутньому, якщо буде потрібно зробити одну зміну в

функціональності класу, її буде зроблено в класі, який реалізує цю одну відповідальність. Коли потрібно робити зміни в класі, який має більше ніж одну відповідальність — це може вплинути на іншу функціональність системи в цілому.

### 2.1.4 Принцип відділення інтерфейсу (Interface Segregation Principle)

Користувачі не повинні бути залежними від інтерфейсів, які вони не використовують. Цей принцип вчить слідкувати за тим, як розробники пишуть свої інтерфейси. Під час написання, розробники повинні подбати про додавання тільки тих методів, які там повинні бути. Якщо було додано методи, які не повинні бути там, тоді класи, що реалізують інтерфейс будуть реалізовувати зайві методи так само як і інші методи. Наприклад, якщо ми створюємо інтерфейс, званий Worker і додаємо метод lunchBreak, тоді всі Workers будуть мати реалізацію цього зайвого методу. А що, якщо дочірній клас не потребує реалізації даного методу та не використовує його? Якщо інтерфейси містять методи, які не є специфічними для них, то вони призводять до того, що інтерфейси називають забрудненими або переповненими. В контексті розробки хорошої програмної архітектури слід уникати створення таких інтерфейсів.

### 2.1.5 Принцип інверсії залежностей (Dependency Inversion Principle)

Залежності всередині системи будуються на основі абстракцій. Модулі верхнього рівня не залежать від модулів нижнього рівня. Абстракції не залежать від реалізацій. Даний принцип дуже важливий і гідний докладного розгляду. Принцип інверсії залежностей в деталях – у протиставленні поганому дизайну, гарний дизайн архітектури повинен бути гнучким, стійким, і пристосованим до повторного використання. Чим нижче взаємозв'язок компонентів додатка один з одним, тим вища гнучкість і мобільність всієї програми в цілому. Програми, що характеризуються високим коефіцієнтом мобільності, дозволяють застосовувати свої компоненти знову і знову для вирішення однотипних задач. Це веде до зниження дублювання коду. Такі програми складаються з великого набору досить

дрібних компонентів, кожен з яких виконує малу частину роботи, але виконує її якісно. Дрібні компоненти набагато простіше тестувати, реалізовувати і супроводжувати. Якщо дотримуватись принципу інверсії залежностей, то програмний код буде більш пристосований до змін і менше залежатиме від контексту виконання. Вірно і зворотне твердження. Якщо додаток є хорошим прикладом вдалого дизайну архітектури, то він, в тій чи іншій мірі, дотримується принципу інверсії залежностей.

## 2.2 Опис алгоритму побудови карти доріг

Слід розглянути приклад мапи приміщення, яка буде використана для обробки, а саме побудови опорних точок та побудови графу доріг на ній. Мапа представлена зображенням 4000 на 4000 пікселів (рис 1.1) та додатковим файлом з описом характеристик карти типу .yaml (рис 2.1).

```
image: mymap.pgm
resolution: 0.050000
origin: [-100.000000, -100.000000, 0.000000]
negate: 0
occupied_thresh: 0.65
free_thresh: 0.196
```

Рисунок 2.1 — Приклад характеристик мапи

У файлі характеристик мапи вказано кількість метрів, яка припадає на одну клітинку карти (піксель), тобто роздільна здатність карти. Ця властивість є однією з основних для дроблення карти на частини та оптимальне розташування направлень, по яким може рухатись мобільний транспортний робот, і тому дає змогу оцінити куди може заїхати робот і т.д.

На основі цих даних розраховується мінімальний розмір комірки, в яку може помістится робот, не зачіпаючи перешкоди, за наступною формулою:

$$cellSize = \lceil robotSize / mapResolution \rceil \quad (4)$$

де $cellSize$ - розмір комірки;

$robotSize$ - розмір робота;

$mapResolution$- роздільна здатність карти.

Блок-схема роботи розбиття мапи на граф шляхів представлена у Додатку Б.

На першому етапі потрібно пройтись по всій карті та розставити опорні точки рівновіддалено одна від одної на розмір комірки (розраховану за формулою 4), в прохідних зонах мапи. Використовуючи метод сітки, можна нанести на мапу опорні точки майбутнього графу доріг, виконавши перевірки на наявність непрохідних ділянок в кожній з клітинок.

Другим етапом визначаємо наявність перешкод в області кожної з опорних точок та виключаємо ті, в зоні яких є перешкоди. Остаточний результат таких операцій зображений на рис. 2.2.

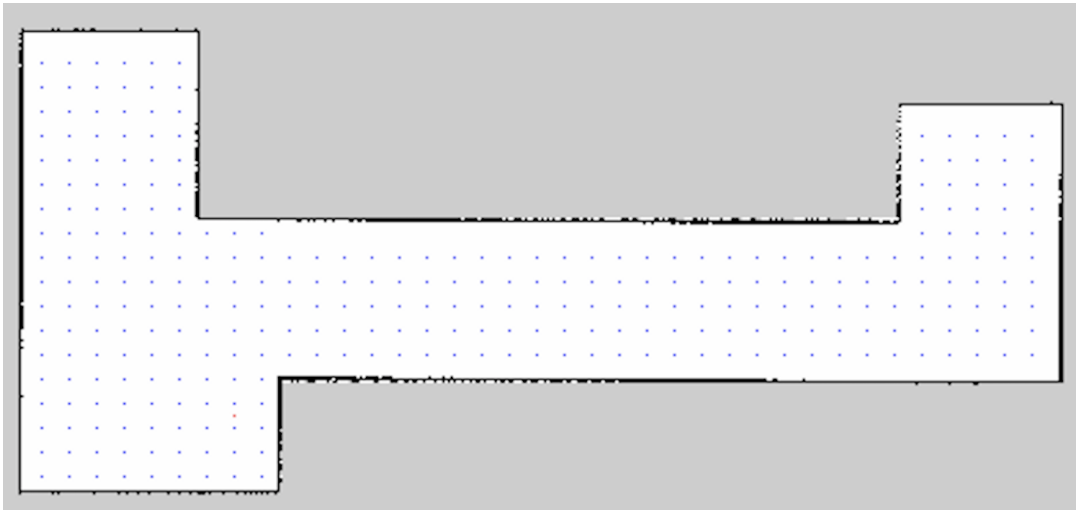

Рисунок 2.2 — Розставлення опорних точок на мапі

Третім етапом потрібно створити граф (мапу доріг) з отриманих опорних точок. З'єднавши кожну опорну точку з її сусідами, отримано результат, який можна побачити на рис. 2.3. Зелена сітка представляє собою можливі направлення руху мобільного робота, зберігаючи відстань від границь приміщення для безпечних поворотів та перевезення вантажу.

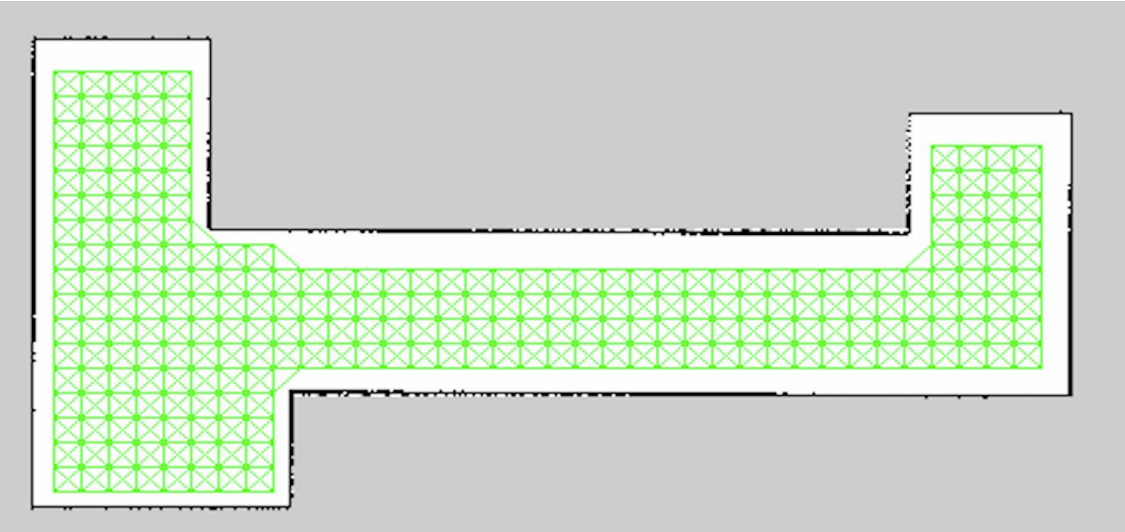

Рисунок 2.3 — Приклад графу доріг

## 2.3 Опис алгоритму побудови оптимального шляху

Оскільки у даній роботі поставлена задача пошуку оптимального шляху до кожної точки виконання операцій, можна знехтувати часом, який робот тратить на виконання операції, а брати до уваги тільки відстань, яку йому потрібно буде проїхати. В цьому випадку оптимальність шляху буде оцінюватись за критерієм оптимальності – довжині шляху, тому дана задача є задачею пошуку найкоротшого маршруту.

Блок-схема роботи алгоритму підсистеми пошуку шляху наведено у Додатку В.

У даній роботі використовується модифікований алгоритм A* з функцією оцінки вартості шляху Euclide для планування шляху до цільової точки, де алгоритм використовує граф шляхів та поточну інформацію про місцезнаходження робота. Обґрунтування вибору модифікації Euclide виконано у розділі 4, а також виконано порівняння ефективності роботи з іншими модифікаціями функцій оцінки вартості шляху. Задача управління задається у вигляді пошуку маршруту до цільової точки на карті.

Спочатку потрібно розглянути поведінку алгоритму для графа шляхів, щоб відпрацювати загальну концепцію роботи алгоритму, після цього застосувати оптимізацію для частини, яка відповідає за побудову графу доступних маршрутів

для пересування. Слід зазначити, що точки виконання типових операцій (завантаження, розвантаження та очікування) наперед включені в граф та виступають одними з його вершин.

Алгоритм для пошуку оптимального шляху був розроблений та протестований на 2D карті перед тим, як застосовувати його для плагіна в 3D симуляції.

Щоб розглянути роботу алгоритму на прикладі необхідно завантажити карту (приклад мапи з перешкодами показаний на рис. 2.4) у розроблений додаток за допомогою кнопки OPEN MAP, обрати координати робота на карті, крок побудови точок графу та натиснути кнопку GENERATE (рис. 2.5). У цьому додатку координати робота задаються користувачем, проте плагін для 3D симуляції буде отримувати актуальні координати мобільного робота за допомогою спеціальної бібліотеки.

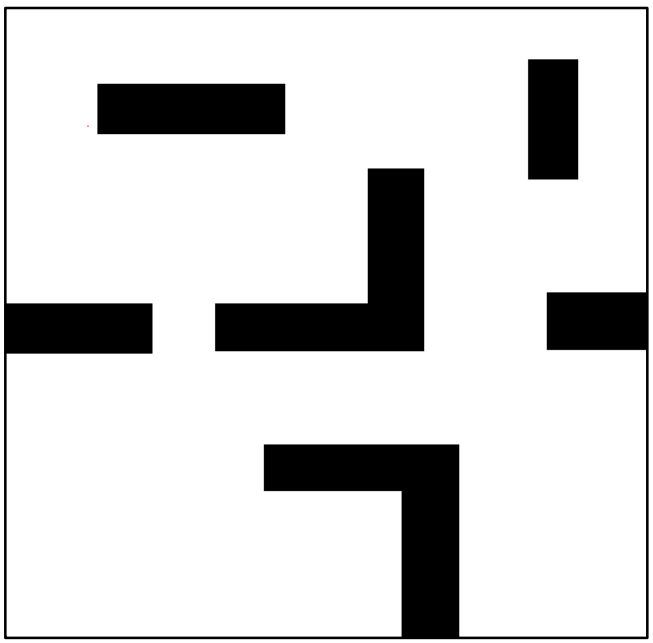

Рисунок 2.4 — Приклад мапи приміщення з перешкодами

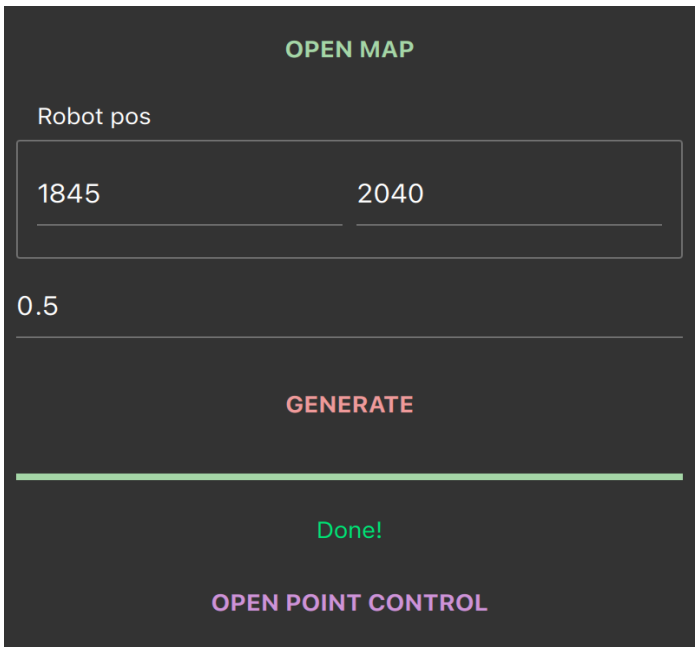


Рисунок 2.5 — Вікно додатка для завантаження мапи

Далі обробляються перешкоди на мапі, виключаються недосяжні ділянки та виставляються опорні точки. Отримано мапу, розбиту на опорні точки (рис. 2.6).

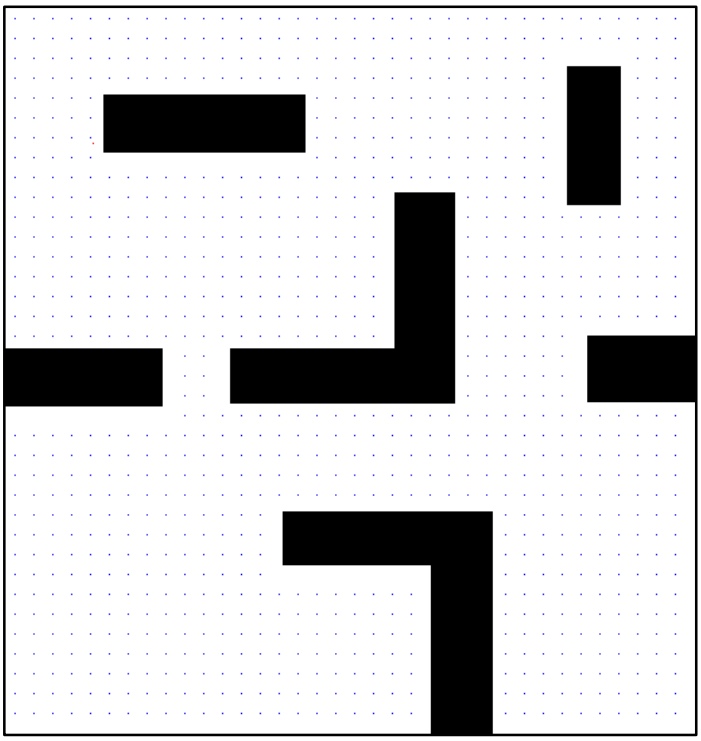

Рисунок 2.6 — Приклад мапи приміщення, розбитої на опорні точки

Опорні точки використовуються як проміжні цільові точки, послідовно досягаючи які, робот рухається до кінцевої мети. Умова близькості опорних точок

дозволяє використовувати лише фрагмент карти – поточні дані сканування, що дозволяє здійснювати процес планування шляху в режимі реального часу. Також опорні точки дозволяють виключити недосяжні ділянки з графу шляхів для безпечного пересування МТР.

Далі проводиться побудова графу шляхів, по якому будуть будуватися маршрути для руху робота (рис. 2.7).

Наступним кроком є формування маршруту. Суть розробленого алгоритму для поставленої задачі у пункті 1.5 складається з того, щоб замість пошуку повноцінного найкоротшого шляху від початкової точки до кінцевої з урахуванням черги потрібних технологічних позицій, проводити поступове розбиття усього шляху на найкоротші відрізки від одної технологічної позиції до іншої відповідно до списку, в якому вони задані. У додатку для цього у вікні вибору технологічних позицій (рис. 2.8) потрібно обрати, які точки (технологічні позиції) робот повинен відвідати.

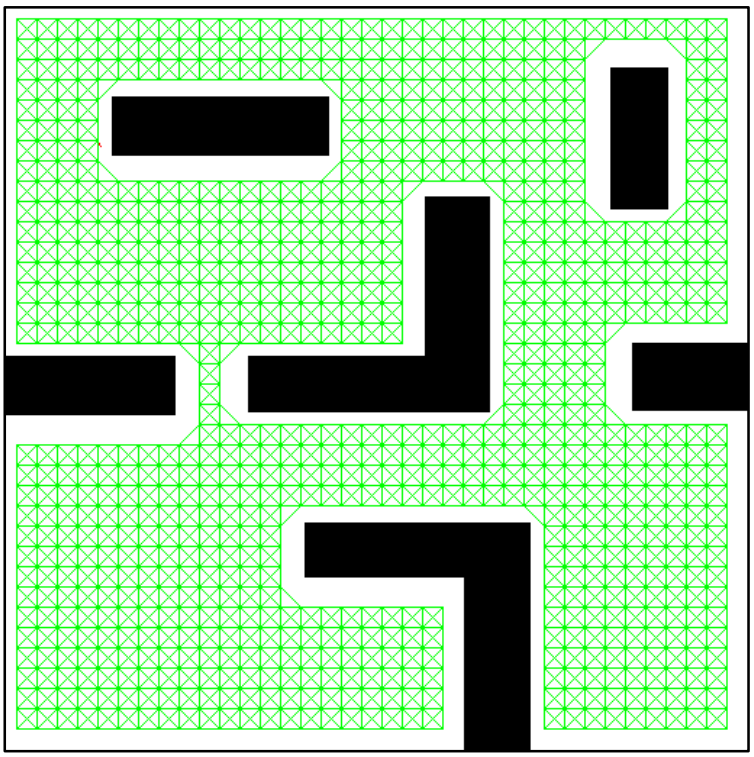

Рисунок 2.7 — Приклад графу шляхів

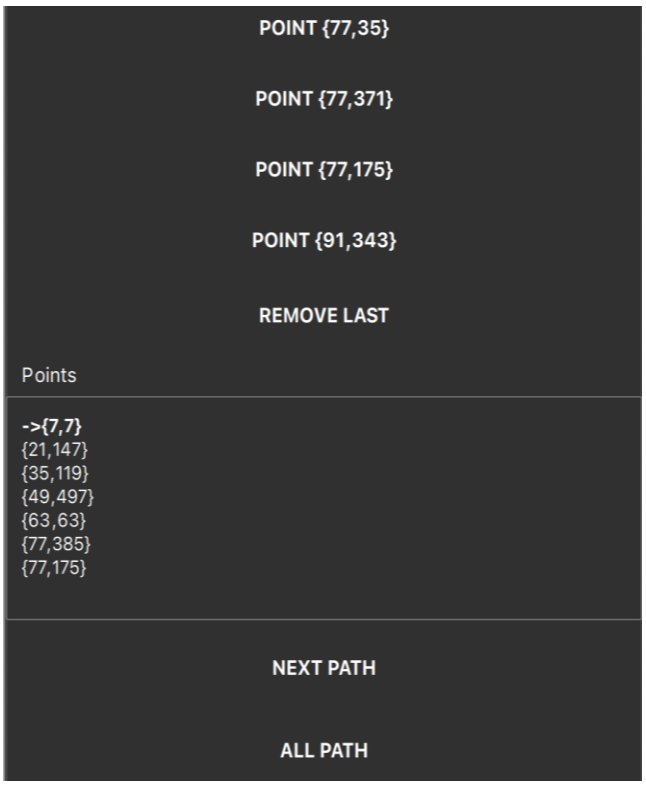


Рисунок 2.8 — Вікно додатку для вибору технологічних позицій

Для тестування обрано, що маршрут робота буде складатися з виконання завдань в 4-х ключових точках та повернення на початкову позицію. Алгоритм окремо послідовно розраховує, який шлях буде найкоротшим від точки до точки, та після цього збирає їх в один маршрут. Перший відрізок шляху зображено на рис. 2.9, де зелена точка — це початкова позиція робота, а червона — перша технологічна позиція як місце призначення.

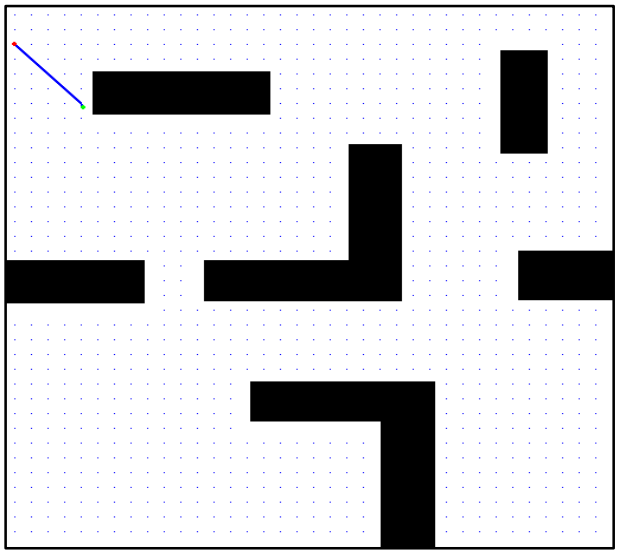

Рисунок 2.9 — Побудова шляху до першої технологічної позиції

Місце призначення попереднього відрізка є початком наступного як показано на рис. 2.10 - 2.12.

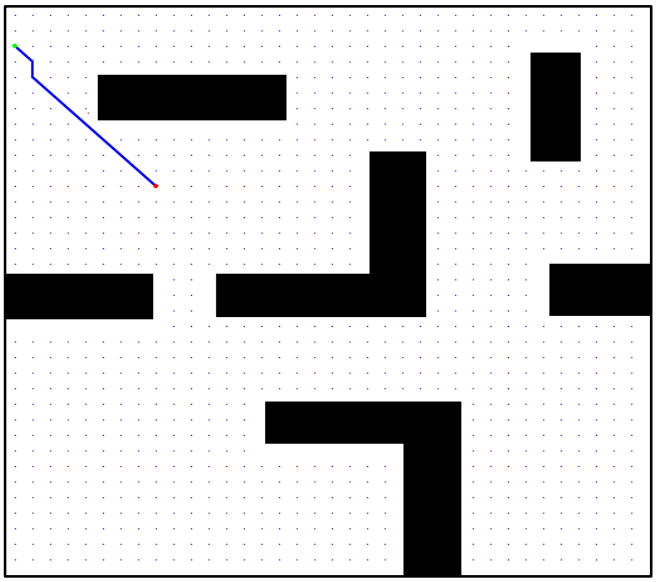

Рисунок 2.10 — Побудова шляху до другої технологічної позиції

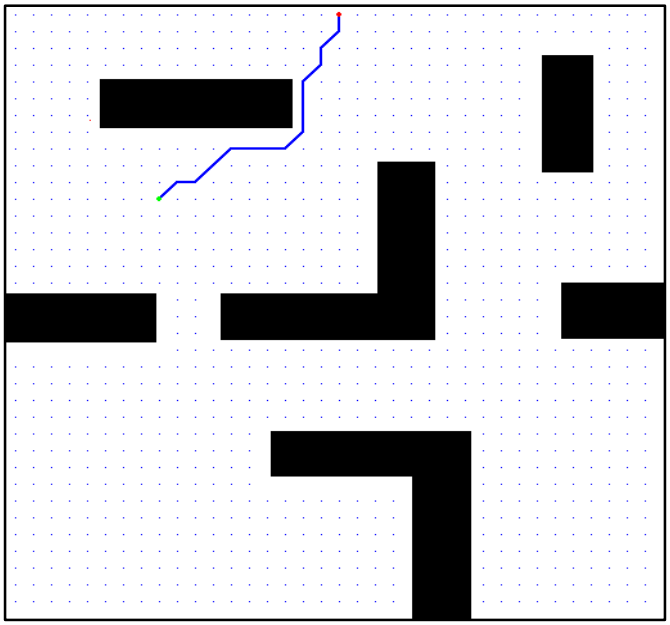

Рисунок 2.11 — Побудова шляху до третьої технологічної позиції

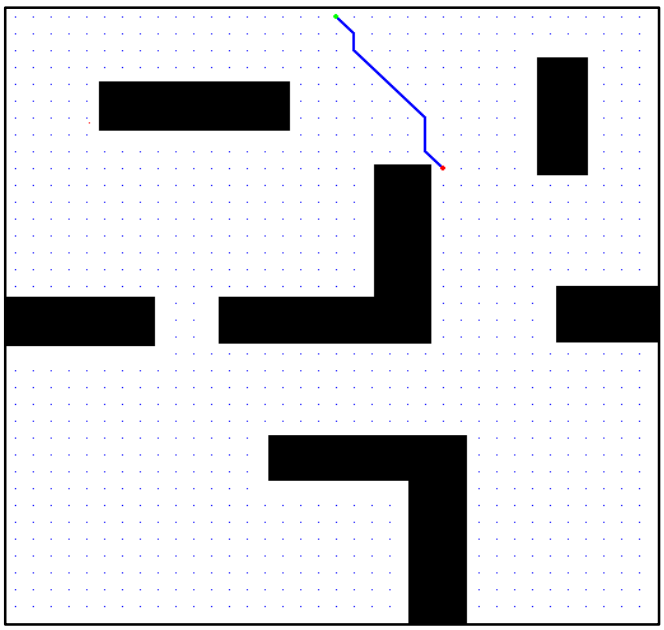

Рисунок 2.12 — Побудова шляху до четвертої технологічної позиції

Після розбиття шляху на відрізки, програмна реалізація відображає весь маршрут робота з позначенням усіх відвіданих ключових технологічних позицій (рис. 2.13).

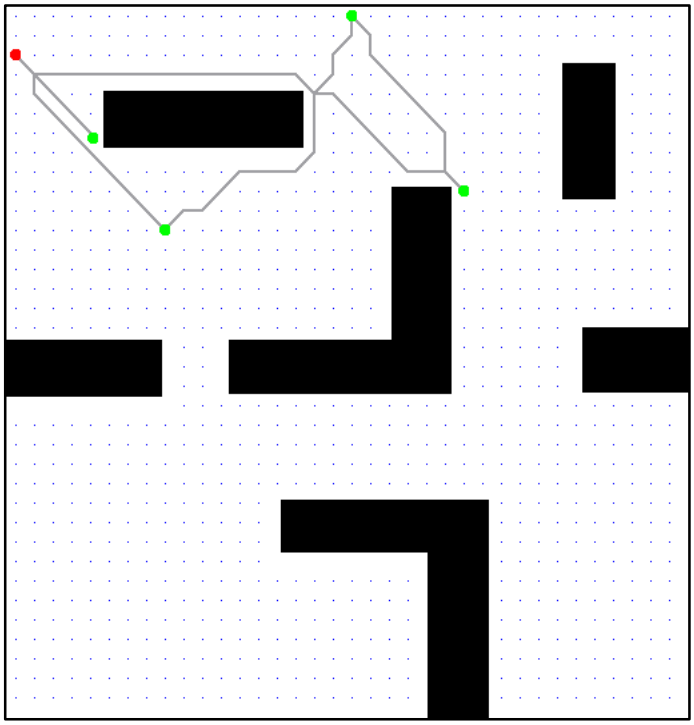

Рисунок 2.13 — Загальний маршрут робота з відвідуванням усіх заданих технологічних позицій

## 2.4 Взаємодія системи з моделлю робота

Робот не може досліджувати невідоме середовище, якщо він не забезпечений потрібними датчиками для отримання інформації про середовище. Різні типи датчиків, таких як сонари, одометри, лазерні далекоміри, інерціальні вимірювальні одиниці (IMU), глобальна система позиціонування (GPS), камери тощо, використовуються для створення робота, здатного відчувати широкий діапазон середовища. Мапа середовища, а саме приміщення, є основною потребою МТР для виконання внутрішніх послуг, таких як переміщення, очікування завантаження вантажів з одного місця та перевезення його в інше. Щоб виконувати такі види послуг, МР повинен не тільки знати про навколишнє середовище, а й під час руху також знати про своє власне розташування в цьому середовищі. Крім того, правильне представлення самого робота в навколишньому середовищі відіграє важливу роль для вирішення багатьох проблем, пов’язаних з автоматичною навігацією.

Оскільки дана підсистема має на меті управління МР, їй потрібні інтерфейси для взаємодії та управління. Такі інтерфейси надає метаопераційна система Robot operating system (ROS) та набір допоміжних засобів таких як: Gmapping, move_base, TF та AMCL.

### 2.4.1 Система ROS

ROS — це гнучка платформа (фреймворк) для розробки програмного забезпечення роботів. Це набір різноманітних інструментів, бібліотек і певних правил, метою яких є спрощення завдань розробки програмного забезпечення для роботів. Хоча ROS — це не операційна система, а сукупність програмних платформ для розробки програмного забезпечення роботів, вона забезпечує апаратну абстракцію, пропонує драйвери пристроїв, бібліотеки, візуалізатори, обмін повідомленнями, менеджери пакетів і багато іншого. ROS випускається відповідно до умов BSD ліцензії та з відкритим вихідним кодом. Незважаючи на важливість оперативності та низьку затримку в управлінні роботами, сама ROS не є

операційною системою реального часу (RTOS). Однак є можливість інтегрувати ROS з системою реального часу. Відсутність підтримки систем реального часу було вирішено під час створення ROS 2, великої редакції API ROS, яка використовуватиме переваги сучасних бібліотек і технологій [17].

Модель робота в ROS містить пакети для моделювання різних аспектів, вказаних у форматі опису робота XML Robot Description Format. Базовий пакет – це стек URDF, який аналізує файли URDF та створює об'єктну модель.

Unified Robot Description Format (Уніфікований формат опису робота URDF) – це специфікація XML для опису моделі робота. За допомогою URDF можна надати такі можливості:

- кінематичний та динамічний опис робота;
- візуальна модель робота;
- моделювання фізичних процесів робота.

Xacro (XML-макроси) – мова макросів формату XML. З xacro можливо створювати короткі XML-файли. Для спрощення файлу URDF, можна використовувати xacro разом із URDF. При цьому слід викликати додаткову програму – аналізатор, який перетворює xacro на URDF.

Пакет move_base надає інтерфейс ROS для налаштування, запуску та взаємодії з навігаційним стеком робота. Він забезпечує реалізацію руху, яка, за умови наявності точки призначення, намагатиметься досягти її за допомогою мобільної бази. Нода move_base об’єднує глобальний і локальний планувальники шляху для виконання його глобального завдання навігації. Він підтримує будь-який глобальний планувальник, який дотримується інтерфейсу nav_core::BaseGlobalPlanner, зазначеного в пакеті nav_core, і будь-який локальний планувальник, який дотримується інтерфейсу nav_core::BaseLocalPlanner, зазначеного в пакеті nav_core.

Запуск ноди move_base на мобільному роботі, який правильно налаштований призводить до того, що робот намагатиметься досягти цілі в межах визначеного користувачем допуску. За відсутності динамічних перешкод move_base пройде в межах цього допуску до цілі. Нода move_base може за бажанням надає можливість

відновлення робота в початковій позиції, коли робот не може проїхати перешкоду та повідомляє користувача про застрягання [30].

tf — це пакет, який дозволяє користувачеві відстежувати кілька координатних кадрів з часом. tf підтримує зв'язок між кадрами координат у деревоподібній структурі, буферизованій у часі, і дозволяє користувачеві змінювати точки, вектори тощо між будь-якими двома кадрами координат у будь-який потрібний момент часу [27].

### 2.4.2 Локалізація робота

Локалізація — це процес пошуку пози робота в навколишньому середовищі. Це, мабуть, найважливіша властивість, якою повинен володіти мобільний робот. Це пояснюється тим, що МР повинен знати свою позицію в навколишньому середовищі, перш ніж він зможе знайти оптимальний маршрут до мети або рухатися за запланованим шляхом до мети.

Більшість мобільних роботів локалізують свою позицію щодо заданої карти на основі показань одометрії. На жаль, заноси коліс викликають додаткові помилки локалізації. Такі помилки призводять до того, що мобільний робот втрачає власні координати і, отже, втрачає здатність автономно переміщатися з однієї заданої точки на мапі до іншої. Рішення проблеми локалізації полягає у використанні інформації про навколишнє середовище від додаткових датчиків. Прикладами використовуваних датчиків є лазерний далекомір і гідролокатор, які вимірюють відстань між роботом і найближчими перешкодами в навколишньому середовищі. Розширений фільтр Калмана та фільтр частинок — це два алгоритми локалізації, які використовують одометрію та додаткові сенсорні дані середовища для локалізації мобільного робота. Обидва алгоритми є імовірнісними методами, які дозволяють принципово врахувати невизначеність оцінки позиції робота та показань датчика.

Проблема локалізації є надзвичайно важливою в реальному світі, оскільки це дає нам імовірнісну оцінку поточного положення та орієнтації робота. Таким

чином, цілком очевидно, що без цих знань робот не зможе приймати ефективні рішення та шукати оптимальні маршрути руху, якщо він не знає, де він знаходиться у світі. Є 3 різні типи проблем з локалізацією.

### 2.4.3 Локальна локалізація

Локальна локалізація — це найпростіша проблема локалізації. Вона також відома як відстеження позиції. У цій задачі робот знає свою початкову позицію, і проблема локалізації тягне за собою оцінку позиції робота, коли він рухається в навколишньому середовищі. Ця проблема не є тривіальною, оскільки в русі робота завжди існує певна невизначеність. Однак невизначеність обмежена регіонами, що оточують робота.

### 2.4.4 Глобальна локалізація

Це більш складна проблема локалізації. У цьому випадку початкова позиція робота невідома, і робот повинен визначити своє розташування відносно наземної карти. Величина невизначеності набагато вище.

### 2.4.5 Проблема “викраденого” робота

Це найскладніша проблема локалізації. Це так само, як проблема глобальної локалізації, за винятком того, що робот може бути “викрадений” в будь-який момент і переміщений в нове місце на мапі.

### 2.4.6 Алгоритми локалізації

Для проблеми локалізації доступний широкий спектр алгоритмів, починаючи від локалізації Монте-Карло, розширеного фільтра Калмана до Маркова і, нарешті, локалізації сітки. Алгоритм локалізації Монте-Карло або MCL є найпопулярнішим алгоритмом локалізації в робототехніці. Після того, як MCL розгорнуто, робот буде

переміщатися всередині своєї відомої карти та збирати сенсорну інформацію за допомогою камери та/або датчиків далекоміра. MCL використовуватиме ці вимірювання датчика, щоб відстежувати позицію робота. MCL часто називають локалізацією фільтра частинок, оскільки він використовує частинки для локалізації робота. Ці частинки є віртуальними елементами, які припускають місцезнаходження робота в даний момент часу. Кожна частинка має позицію та орієнтацію, і це означає, що алгоритм надає дані, де може бути розташований робот. Ці частинки повторно відбираються щоразу, коли робот рухається та "відчуває" навколишнє середовище. Для цього проєкту була використана модифікована версія цього алгоритму, відома як «Адаптивна локалізація Монте-Карло», оскільки цей модифікований алгоритм динамічно коригує кількість частинок протягом певного періоду часу, коли робот переміщається по карті, що робить процес більш ефективним. На рис. 2.14 частинки локалізації зображені як червоні стрілки.

Такий підхід дозволяє роботу автономно уникати як статичних, так і динамічних перешкод. Щоб навігація працювала добре, дуже важливо мати мапу навколишнього середовища. MP може працювати в середовищі, для якого йому ще не надали мапу заздалегідь. Проте в такому випадку він повинен спочатку створити її. Процес створення карти називається мапування. Процес сканування приміщення та зберігання мапи відбувається за допомогою технології Gmapping.

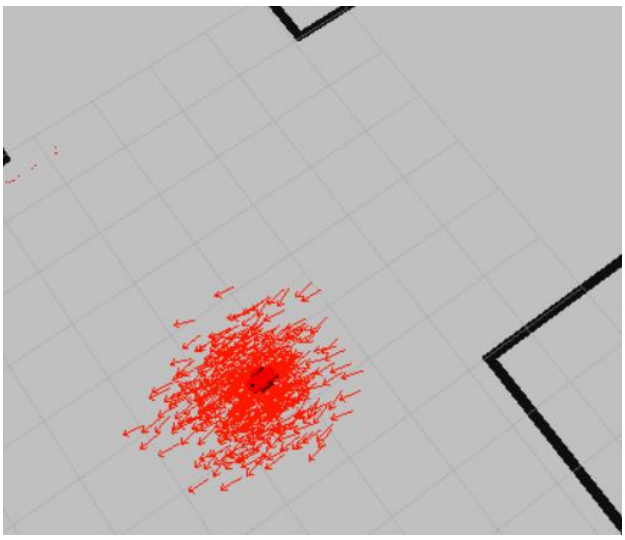

Рисунок 2.14 — Процес роботи алгоритму локалізації AMCL

Gmapping — це реалізація конкретного алгоритму SLAM (Simultaneous Localization And Mapping). SLAM — це одночасне визначення розташування робота та побудови мапи. Це завдання пов'язане з побудовою мапи невідомого середовища мобільним роботом під час навігації по мапі, що будується. Вихідними даними є деяке середовище та робот із датчиками. В кожний момент часу необхідно знати розташування робота і мапу тієї частини навколишнього простору, яка була доступна для спостереження з початкового часу до поточного. Розташування робота представляється в системі локальних координат, пов'язаних з вихідним становищем робота, оскільки інформації про його положення щодо простору немає [31]. Пакет gmapping містить вузол ROS slam_gmapping, який будує 2D-карту, використовуючи дані вимірювань лазерного датчика та дані про положення мобільного робота. Нода slam_gmapping підписана на топіки /scan і /tf, щоб отримати необхідні дані для побудови карти. Цей вузол зчитує дані з лазера та перетворень, а потім створює OGM (Occupancy Grid Map) на основі цих даних. Карта, що створюється, публікується на топік /map протягом усього процесу відображення. Топік /map використовує повідомлення типу nav_msgs/OccupancyGrid. В OGM зайнятість представлена цілим значенням у діапазоні від 0 до 100. Значення 0 означає повністю незайняту ділянку, 100 означає повністю зайняту (наприклад, стіна), а спеціальне значення -1, що означає невідомий статус ділянки [28].

## 2.5 Взаємодія плагіна з системою ROS

Система ROS побудована за принципом нодів, які можуть передавати між собою інформацію. Основним механізмом, який використовується вузлами ROS для зв’язку, є надсилання та отримання повідомлень. Повідомлення організовано за певними категоріями, які називаються топіками. Ноди можуть публікувати повідомлення на певний топік або підписатися на топік, щоб отримувати інформацію.

До таких нодів відносяться й ті, які описують приміщення, ноди МР, якими буде відбуватися безпосередньо керування, ноди датчиків МР, ноди мапи приміщення та поточної позиції МР на ній.

Плагін, який буде розроблятися, також буде використовувати дану систему взаємодії. Приклад взаємодії вузлів для плагіну у системі ROS наведено у Додатку А. Власне плагін пошуку шляху на мапі повинен синхронізувати дані у 3D симулятор, тому його розробка складається з трьох підзадач:

1. обробка мапи (отримання мапи 2D приміщення, розбиття його на опорні точки, побудова графу шляхів та надсилання цих даних на певний топік);
2. визначення технологічних позицій (розставлення міток технологічних позицій та надсилання їх координат на певний топік);
3. пошук та відображення шляху до певної технологічної позиції (визначення позиції робота на карті, підписання на топіки отримання графу шляхів та координат технологічних позицій, побудова та візуалізація шляху).

Загалом структура потоку та обміну даними в системі що розробляється, а також всі її компоненти разом з їх зв’язками наведено у Додатку Ж.

Результатом першого етапу розробки є граф шляхів. Головна реалізація цього пункту, що включає пошук недосяжних ділянок та їх виключення, розставлення опорних точок та побудова графу шляхів представлена на рис. 2.15 - 2.16.

```
1  void PathGraph::generateNodes() {
2      nodes.clear();
3
4      nodesImage = rawImage;
5      nodes.reserve(1000);
6
7      for (int i = 0; i < nodesImage.width(); i += gridSize) {
8          for (int j = 0; j < nodesImage.height(); j += gridSize) {
9
10             int x = i + gridSize / 2;
11             int y = j + gridSize / 2;
12
13             auto cPixel = nodesImage.pixelColor(x, y).rgb();
14
15             if(cPixel != wallColor.rgb() && cPixel != outColor.rgb()) {
16                 // nodesImage.setPixelColor({x, y}, nodeColor); // remove later
17                 nodes.insert(QPointF(x, y));
18             }
19         }
20     }
```

Рисунок 2.15 — Програмний код процесу розбиття мапи на опорні точки

Результатом другого етапу розробки є надсилання в топік координат виставлених технологічних позицій. Головна реалізація цього пункту представлена на рис. 2.17.

Результатом третього етапу розробки є побудова знайденого оптимального шляху та рух по ньому, які реалізовані на рис. 2.18 - 2.19.

```
    QSet<QPointF> nodesToRemove;

    for (auto &node : (nodes)) {

        bool canReach = true;

        int cellX = node.x() - gridSize / 2;
        int cellY = node.y() - gridSize / 2;

        for (int i = cellX; i < cellX + gridSize + 1; ++i) {
            for (int j = cellY; j < cellY + gridSize + 1; ++j) {
                if(rawImage.pixelColor({i,j}).rgb() == wallColor.rgb()){
                    canReach = false;
                    break;
                }
            }
        }

        if(!canReach) nodesToRemove.insert(node);
    }

    nodesImage = rawImage;
    nodes = nodes.subtract(nodesToRemove);

    for (auto &node : (nodes))
        nodesImage.setPixelColor({int(node.x()), int(node.y())}, nodeColor);
}
```

Рисунок 2.16 — Програмний код визначення недосяжних точок та виключення їх

```
void TechPosition::publishPositions() {
  if( ros::ok() && tech_position_publisher_ )
  {
    std_msgs::String msg;

    QJsonObject obj;
    QJsonArray arr;

    int num = 0;

    for (auto pos : position_nodes_) {
      QJsonObject _obj;

      _obj.insert("pos_name", QString("TP #%1").arg(num++));

      auto vec3 = pos->getPosition();

      _obj.insert("x", vec3.x);
      _obj.insert("y", vec3.y);
      _obj.insert("z", vec3.z);

      arr.append(_obj);
    }

    obj.insert("positions", arr);
    QJsonDocument doc(obj);
    msg.data = doc.toJson().toStdString();

    tech_position_publisher_.publish( msg );
  }
}

```

Рисунок 2.17 — Реалізація визначення та надсилання даних про технологічні позиції

```
1  void PosMover::moveToPosition(QVector<double> raw_robotPosAMCL, QVector<double>  raw_pos) {
2      geometry_msgs::PoseWithCovarianceStamped robotPosAMCL;
3      robotPosAMCL.pose.pose.position.x = raw_robotPosAMCL[0];
4      robotPosAMCL.pose.pose.position.y = raw_robotPosAMCL[1];
5      robotPosAMCL.pose.pose.position.z = raw_robotPosAMCL[2];
6
7      Ogre::Vector3 pos;
8
9      pos.x = raw_pos[0];
10     pos.y = raw_pos[1];
11     pos.z = raw_pos[2];
12
13     MoveBaseClient _mbc("move_base", true);
14
15     while(!_mbc.waitForServer(ros::Duration(1.0))) {
16         ROS_INFO("Waiting for move_base server");
17     }
18
19     move_base_msgs::MoveBaseGoal goal;
20
21     Ogre::Vector3 rPos;
22
23     rPos.x = robotPosAMCL.pose.pose.position.x;
24     rPos.y = robotPosAMCL.pose.pose.position.y;
25     rPos.z = robotPosAMCL.pose.pose.position.z;
26
```

Рисунок 2.18 — Взаємодія з move_base для керування роботом

```
27     auto _diff = pos - rPos;
28
29     goal.target_pose.header.frame_id = "base_link";
30     goal.target_pose.header.stamp = ros::Time::now();
31
32     goal.target_pose.pose.position.x = _diff.x;
33     goal.target_pose.pose.position.y = _diff.y;
34     goal.target_pose.pose.position.z = 0.0;
35
36     goal.target_pose.pose.orientation.x = 0.0;
37     goal.target_pose.pose.orientation.y = 0.0;
38     goal.target_pose.pose.orientation.z = 0.0;
39     goal.target_pose.pose.orientation.w = 0.1; // was 1.0
40
41     _mbc.sendGoal(goal);
42     _mbc.waitForResult(ros::Duration(CONFIG::MOVE_TIMER_INTERVAL_MS / 1000));
43
44     if(_mbc.getState() == actionlib::SimpleClientGoalState::SUCCEEDED)
45         Q_EMIT positionReached();
46     else
47         Q_EMIT positionDoesNotReached();
48 }
```

Рисунок 2.19 — Прокладання шляху до точки призначення

Наведені програмні частини являють собою малу долю головних програмних одиниць розробленої системи, які відображають ключові аспекти її роботи. Загальну діаграму класів усіх частин плагіну наведена у Додатку Г. Вона включає в себе всі інструменти, панелі та інші виконавчі модулі системи, а також їх взаємодію в контексті програмного забезпечення.

Загальна кодова база проекта системи пошуку оптимального шляху та всі вихідні файли для проведення симуляції робота та приміщення знаходяться на загальнодоступному ресурсі за посиланням: https://github.com/OlhaKryvych/PathFindingROS

## 2.6 Висновки до розділу

У даному розділі розглянуто принципи проектування архітектури плагіну. Виконано побудову графу шляхів за методом сітки на основі отриманої карти з файлу, з урахуванням особливостей карти, таких як: роздільна здатність, розміри тощо. Було виконано пошук оптимального шляху за алгоритмом A* з модифікацією функції оцінки оптимальності шляху Euclide.

Було створено тестовий додаток для проведення тестування частин реалізованої системи. Проведено тестування даної реалізації у розробленому додатку перед використанням його для плагіну.

Також розглянуто систему взаємодії надсилання та отримання даних у ROS, а саме механізм нодів та топіків, та модель підписок та публікацій між ними, локалізацію робота на мапі та проблеми, які існують наразі при визначенні позиції MP. Проведено огляд технологій, потрібних для роботи плагіну, а саме move_base, gmapping, amcl, tf.

Спроектовано та розроблено плагін для візуалізації маршруту руху у 3D приміщенні, тестування якого буде проведено у Розділі 3.

# 3 ТЕСТУВАННЯ ТА ВІЗУАЛІЗАЦІЯ СТВОРЕНОГО АЛГОРИТМУ

У даному розділі розглянуто програмні середовища для моделювання поведінки робототехнічних систем в умовах реального часу Gazebo та програмний застосунок для візуалізації поведінки роботів RVIZ. Було створено модель невеликого технологічного приміщення наближеного до звичайного складського приміщення з декількома секціями. Було побудовано карту прохідності для створеного приміщення використовуючи технології Gmapping, отриману карту приміщення було вивантажено з ноди сервера карт та збережено для її подальшого використання.

Було протестовано роботу алгоритму розбиття карти та створення графу доріг на основі опорних точок (нодів графа). Було створено канал взаємодії різних модулей системи для наочної візуалізації процесу пошуку шляху, поточного графа доріг та опорних точок поточного графа.

Було створено інтерфейс для плагіну з можливістю створення списку технологічних позицій для обходу, а також інші функціональні можливості, такі як очищення поточного списку для об’їзду, зупинка руху робота та примусове оновлення його позиції.

Було встановлено та протестовано зв’язок симуляції RVIZ з Gazebo для синхронної демонстрації руху робота з моделюванням фізики реального світу, було перевірено правильність подання команд управління системою розрахунку оптимального шляху на виконавчі ноди для подальшого руху робота.

## 3.1 Програмне середовище для моделювання роботи

Gazebo — це 3D-симулятор робототехніки з відкритим кодом. Gazebo був частиною Player Project з 2004 по 2011 рік. У ньому інтегровані фізичний движок ODE, рендеринг зображення використовуючи OpenGL та API для моделювання датчиків і керування приводом. У 2011 році Gazebo стала незалежним проектом за підтримки Willow Garage. У 2012 році Фонд робототехніки з відкритим кодом

(OSRF) став розпорядником проєкту Gazebo. OSRF змінив назву на Open Robotics у 2018 році.

Gazebo може використовувати кілька високопродуктивних фізичних механізмів, таких як ODE, Bullet тощо (за замовчуванням — ODE). Він забезпечує реалістичне зображення середовищ, включаючи високоякісне освітлення, тіні та текстури. Він може моделювати датчики, які «бачать» змодельоване середовище, такі як лазерні далекоміри, камери (у тому числі ширококутні), датчики в стилі Kinect тощо.

ODE — це високопродуктивна бібліотека з відкритим вихідним кодом для моделювання динаміки твердого тіла. Це повнофункціональний, стабільний, завершений і незалежний від платформи простий у використанні інструмент з API на таких мовах як C та C++. Він має вдосконалені типи з'єднань та вбудоване виявлення зіткнень з перешкодами та з урахуванням тертя рухомих об'єктів. ODE корисний для моделювання транспортних засобів, об’єктів у середовищах віртуальної реальності та віртуальних істот. Нині він використовується в багатьох комп'ютерних іграх, інструментах 3D та інструментах моделювання.

Bullet Physics SDK — це професійна бібліотека для виявлення зіткнень з відкритим кодом, динаміки твердих і м’яких тіл, написана мовою програмування C++. Бібліотека в першу чергу призначена для використання в іграх, візуальних ефектах та моделюванні роботи. Бібліотека безкоштовна для комерційного використання за ліцензією zlib.

pybullet — це простий у використанні модуль написаний на Python для моделювання фізики, робототехніки та машинного навчання. За допомогою pybullet можливо завантажувати шарнірні тіла з URDF, SDF та інших форматів файлів. pybullet забезпечує моделювання прямої динаміки, обчислення зворотної динаміки, пряму та зворотну кінематику, виявлення зіткнень і перевірки на перетин променів. Крім моделювання фізики, pybullet підтримує рендеринг з використанням OpenGL та підтримкою гарнітур віртуальної реальності.

RVIZ (скорочено від «ROS visualization») — це програмний інструмент для тривимірної візуалізації для роботів, датчиків і алгоритмів. Це дає змогу побачити

сприйняття роботом свого світу (реального чи змодельованого). Мета RVIZ - дозволити користувачу уявити стан робота. Він використовує дані датчиків, щоб спробувати створити точне зображення того, що відбувається в оточенні робота. Інструмент RVIZ дозволяє в реальному часі відображати на 3D-сцені всі компоненти робототехнічної системи — системи координат, рухомі частини, покази датчиків, зображення з камери [19].

Різницю між RVIZ та Gazebo можна виразити таким чином, що RVIZ показує те, що робот думає, що відбувається, а Gazebo показує стан речей, що відбувається насправді.

## 3.2 Підключення плагіну пошуку оптимального шляху

Розробка плагіну включає в себе підключення додаткових функцій RVIZ, таких як панель (panel) та інструмент (tool). Панель може використовуватися для відправки позиції цілі прямо до системи керування рухом робота. Інструмент - це клас, який визначає, як події миші взаємодіють з візуалізатором. Він буде використовуватися для задання технологічних позиції на мапі.

Обидва ці компоненти гнучко зв’язані між собою, але можуть бути використані й окремо для різних цілей. Інструмент для розміщення технологічних позицій також має змогу візуалізувати ноди графа для спрощення розуміння оператором поточної обстановки на карті приміщення.

Для підключення плагіна для пошуку шляху та візуалізації роботи у Gazebo необхідно виконати послідовність налаштувань. Блок-схема послідовності підключення вказана на рис. 3.1., а візуальне представлення етапів управління наведено у Додатку И.

Першим етапом створюється модель приміщення в Gazebo (рис. 3.2). Файл, який містить усі налаштування моделі, зберігається як формат .world та підключається до проекту.

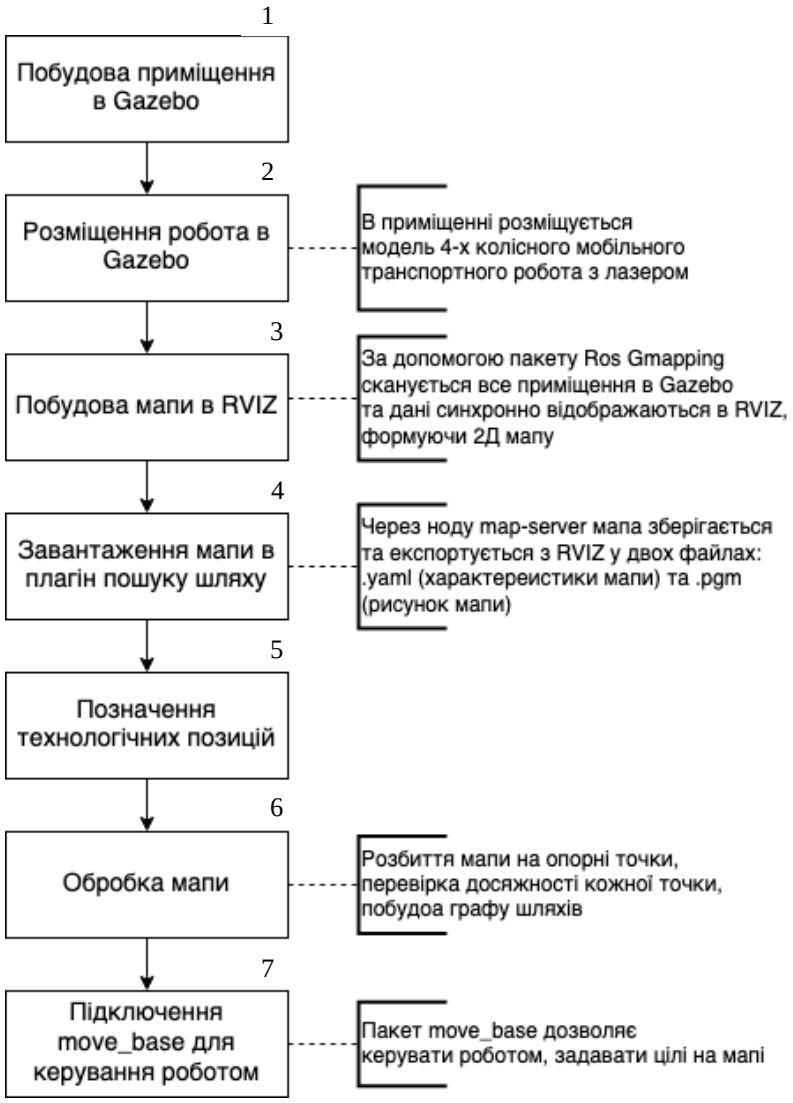


Рисунок 3.1 — Блок-схема підготовки проекту для підключення плагіну

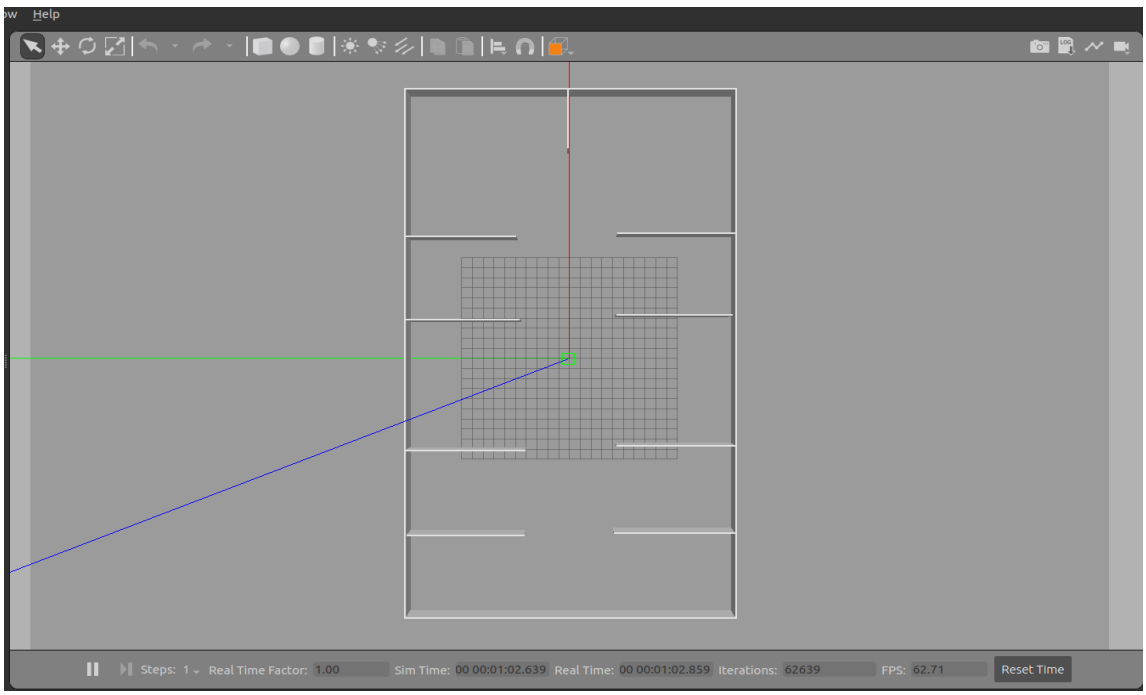


Рисунок 3.2 — Модель промислового приміщення в Gazebo

Додавання мобільного робота у приміщення представлено на рис. 3.3. Для тестування завантажено 4-х колісну модель робота Jackal з лазером hokuyo, зображену на рис. 3.4. Дальність лазера налаштовано на 15 метрів.

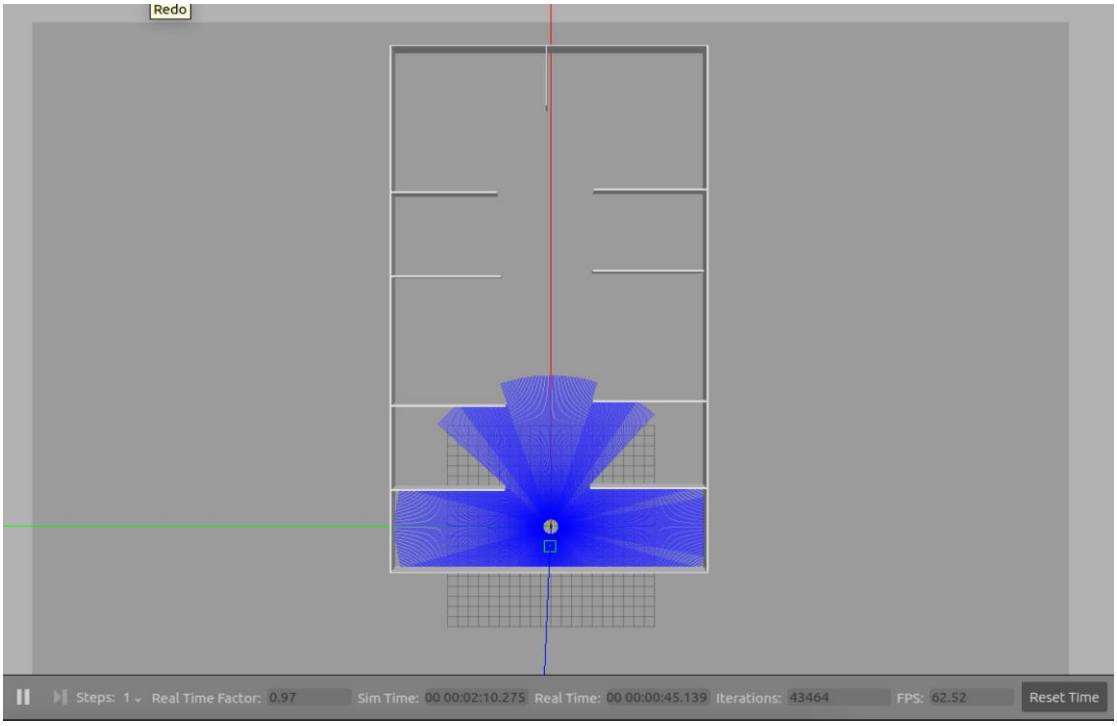


Рисунок 3.3 — Модель промислового приміщення та модель мобільного робота в Gazebo

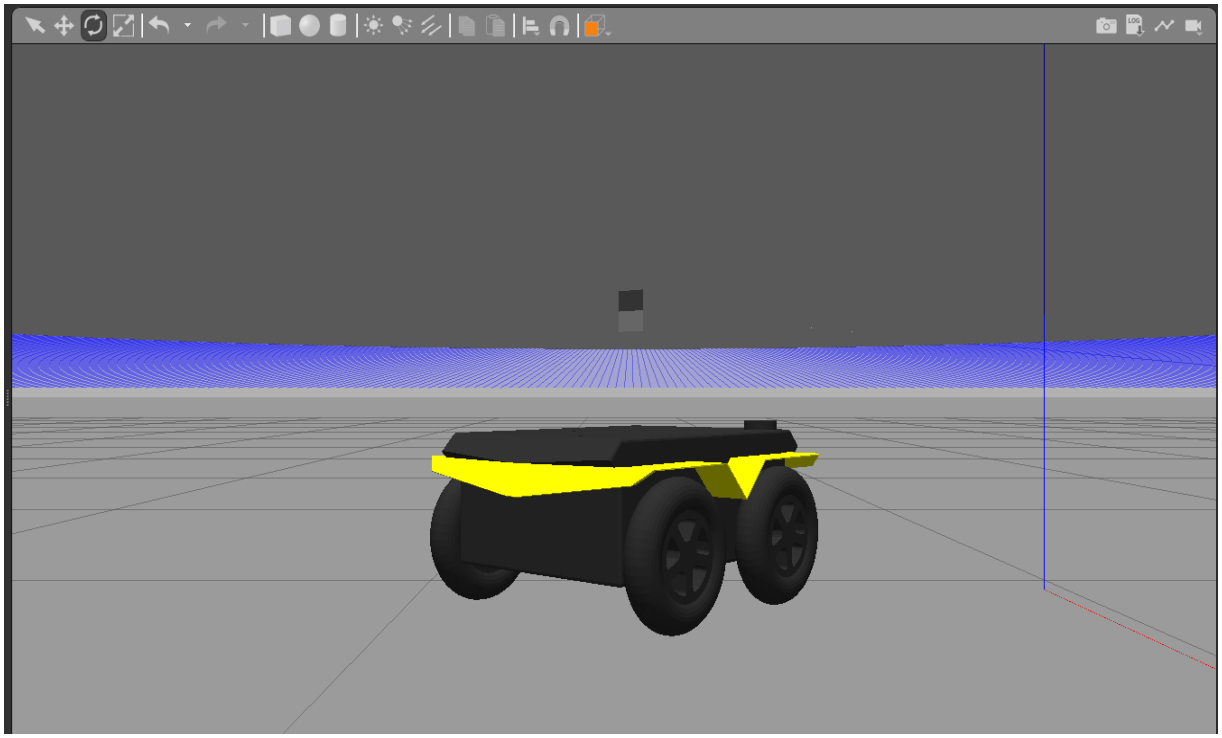

Рисунок 3.4 — Модель мобільного робота Jackal

За допомогою ноди gmapping просканово приміщення в RVIZ, та отриману мапу збережено та експортовано за допомогою ноди map_saver у файли .yaml та .pgm. Отриману мапу підключено по проекту для автоматичного завантаження в RVIZ (рис. 3.5).

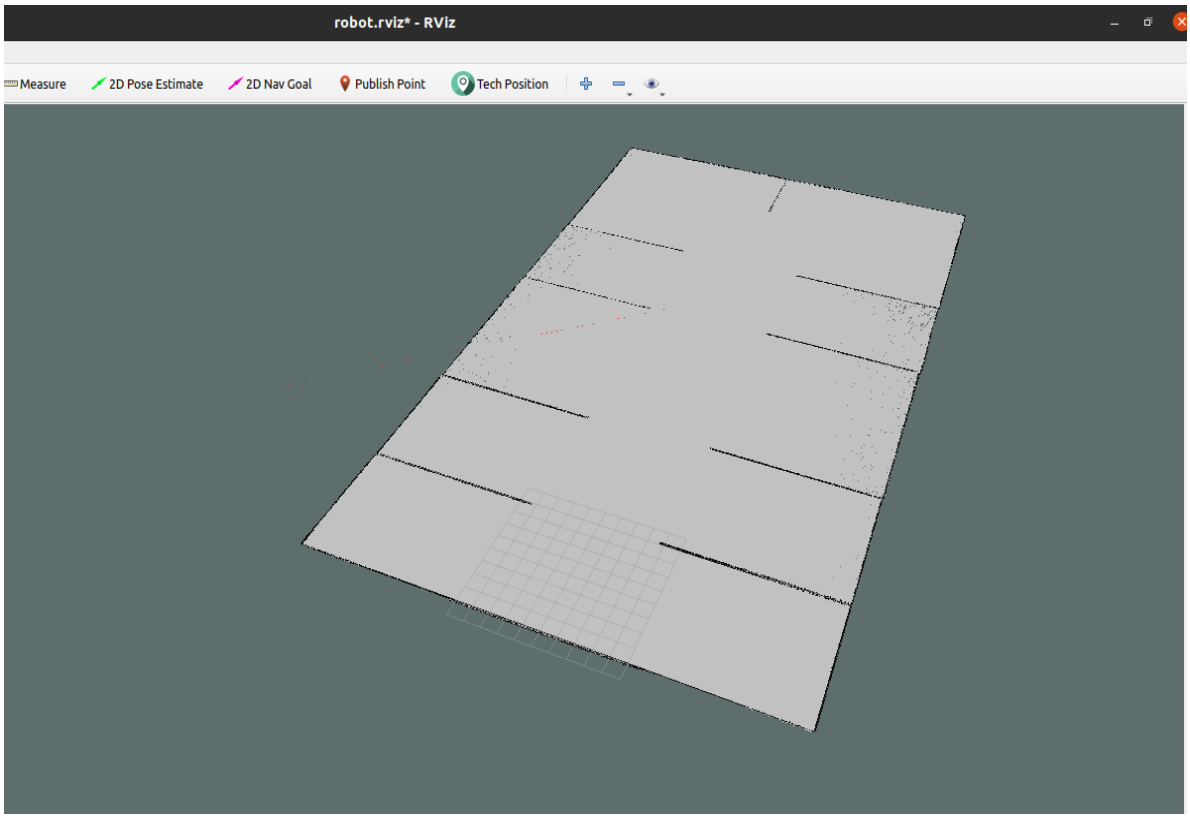


Рисунок 3.5 — Мапа приміщення в середовищі RVIZ

До RVIZ також додано модель робота, лазер, підключено amcl ноду (зображену як сукупність червоних направлених стрілок) для локалізації робота на карті. На лівій панелі на рис. 3.6 зображено підключену першу частину плагіну PathGraph, яка відповідає за обробку мапи. Плагін отримує дані з завантаженого файлу мапи та відображає схему мапи та будує граф шляхів.

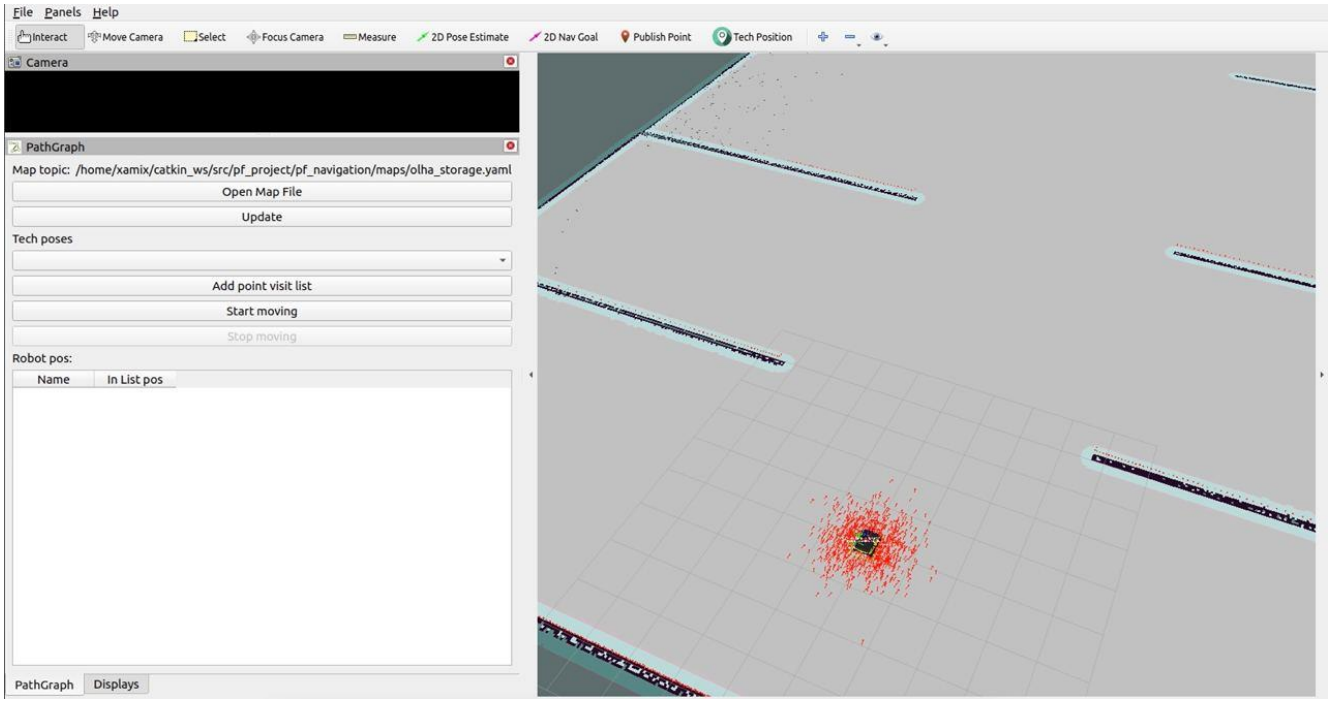


Рисунок 3.6 — Підключена перша частина плагіну PathGraph

Наступним кроком потрібно проставляти та визначати координати ключових технологічних позицій, які робот повинен об’їхати. Для цього розроблено другу частину плагіну пошуку шляху, а саме Tech Position, що розміщена у верхній частині меню RVIZ на рис. 3.7. Дана частина дозволяє розставляти ключові позиції у вигляді зелених квадратів та передає координати цих точок у частину плагіну, яка реалізує пошук оптимального шляху.

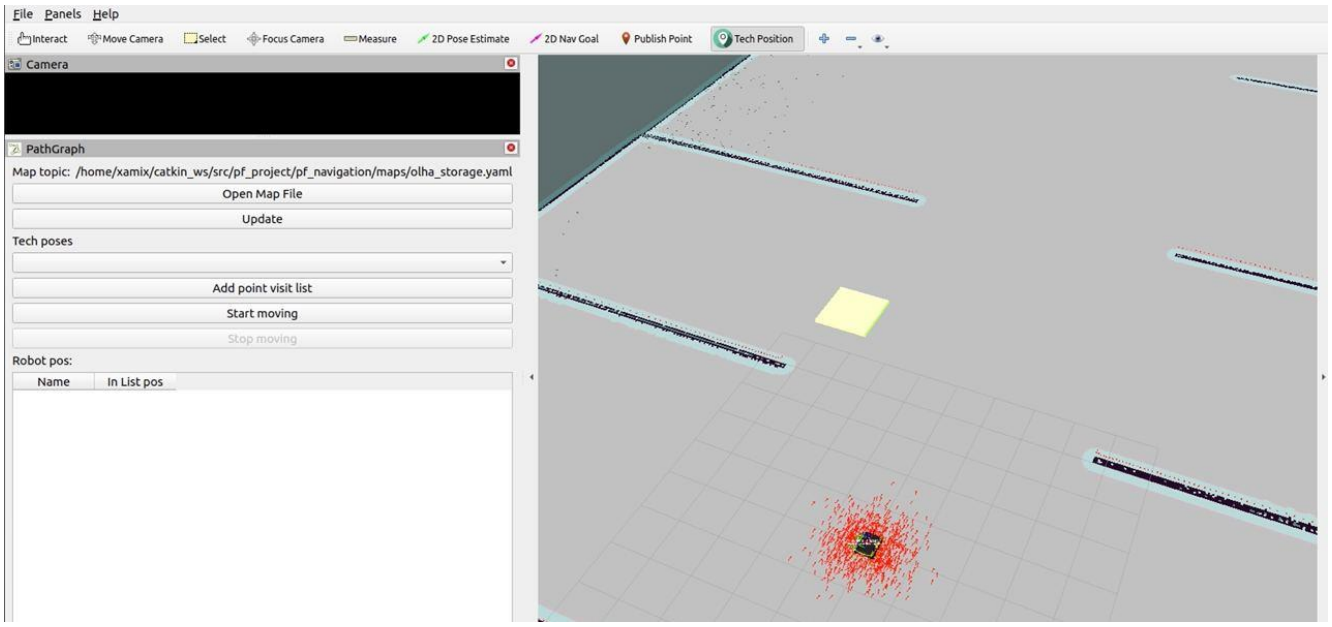


Рисунок 3.7 — Позначення технологічної позиції на мапі

Для того, щоб пересуватися по мапі потрібно підключити пакет move_base. Він надає засоби керування роботом. Для початку руху роботу необхідно вказати початкову точку, де саме на мапі він знаходиться, тобто локалізувати його у визначеній позиції. Це робиться за допомогою функції 2D Pose Estimate. Після цього можна задавати точку призначення за допомогою 2D Nav Goal. Це реалізується вказанням потрібної кінцевої точки на мапі стрілкою рожевого кольору на рис. 3.8. Від робота до стрілки прокладається локальний шлях, у вигляді зеленої лінії (рис. 3.9).

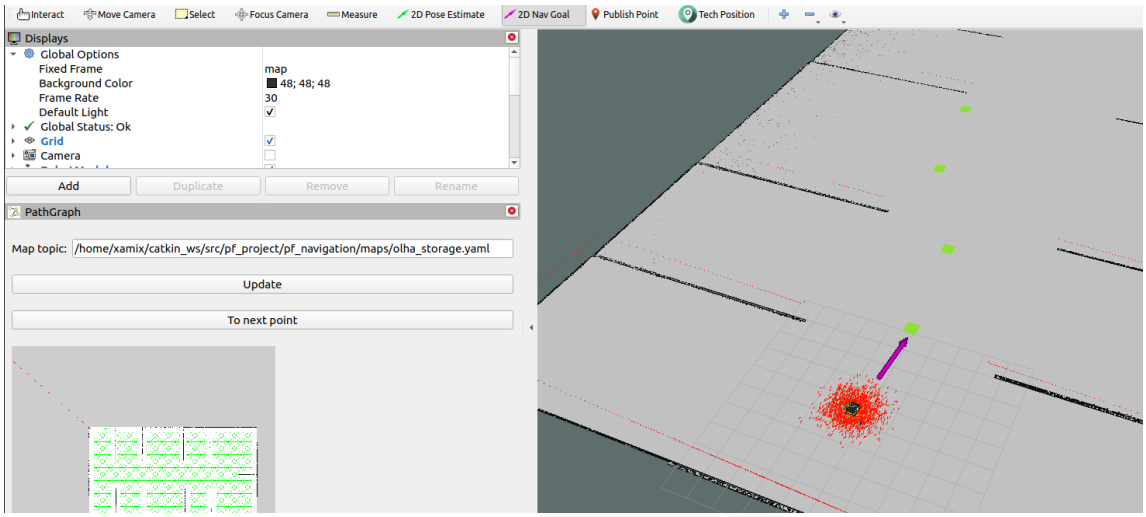


Рисунок 3.8 — Задання цілі для мобільного робота на мапі

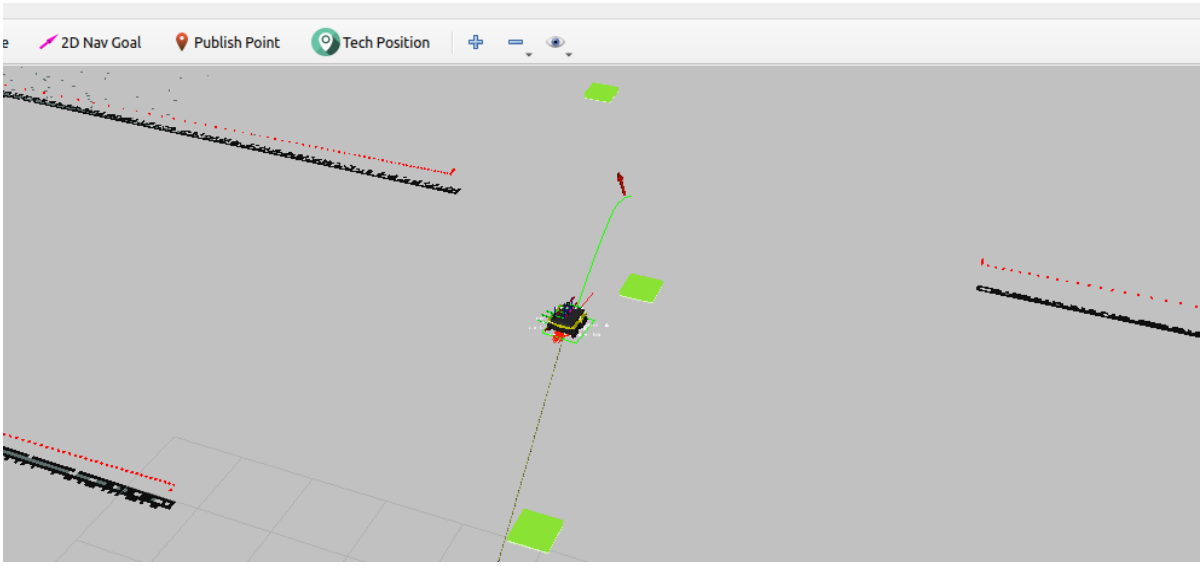


Рисунок 3.9 — Візуалізація шляху до точки, зображене за допомогою move_base

## 3.3 Візуалізація пошуку оптимального маршруту руху

Використовуючи частину плагіну, яка відповідає за побудову графу, можна відобразити граф шляхів, складений з опорних точок на мапі (рис. 3.10). При натисненні на кнопку Update мапа оновлюється та поділяється на ділянки, визначаються опорні точки та зображуються у вигляді рівновіддалених зелених точок. Також визначається стартова позиція відносно мапи.

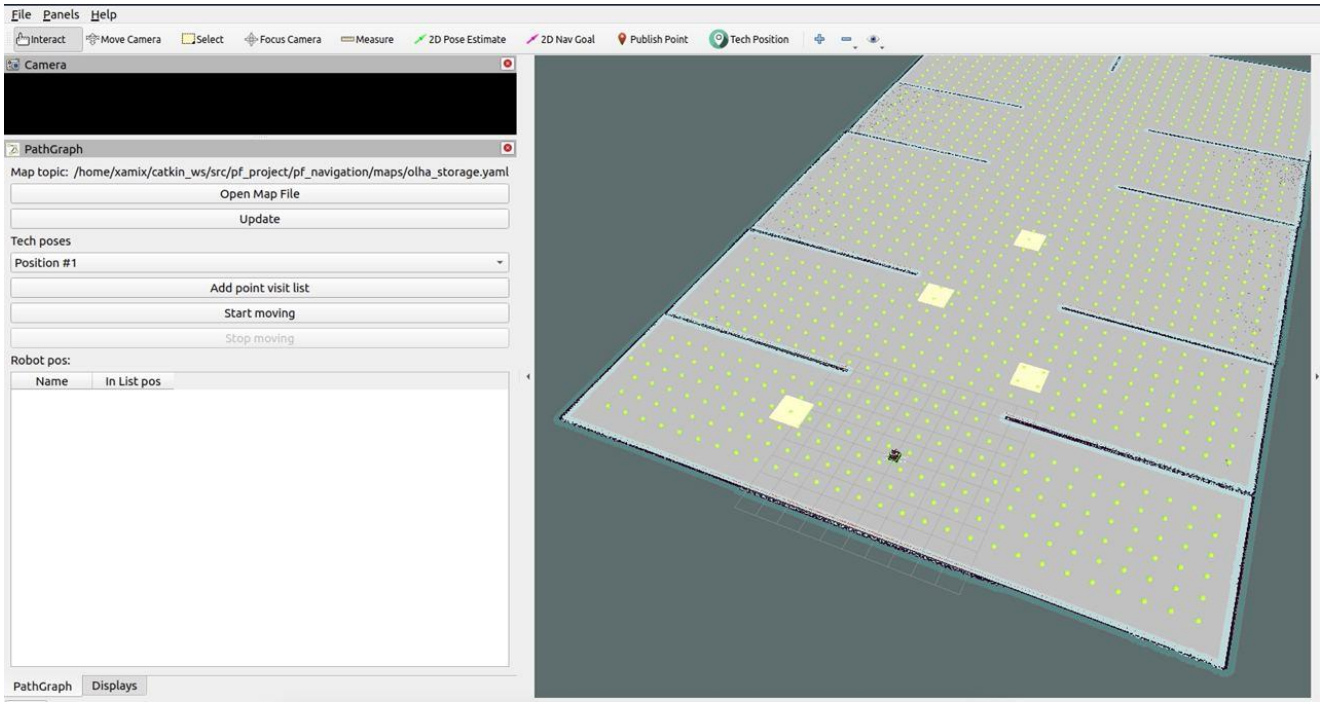


Рисунок 3.10 — Граф шляхів, зображений на мапі з заданими технологічними позиціями

У меню Tech Poses обирається потрібна технологічна позиція на мапі та натискається кнопка Add point visit list. Отримана черга на відвідування з доданих технологічних позицій зображена у списку Robot pos (рис. 3.11).

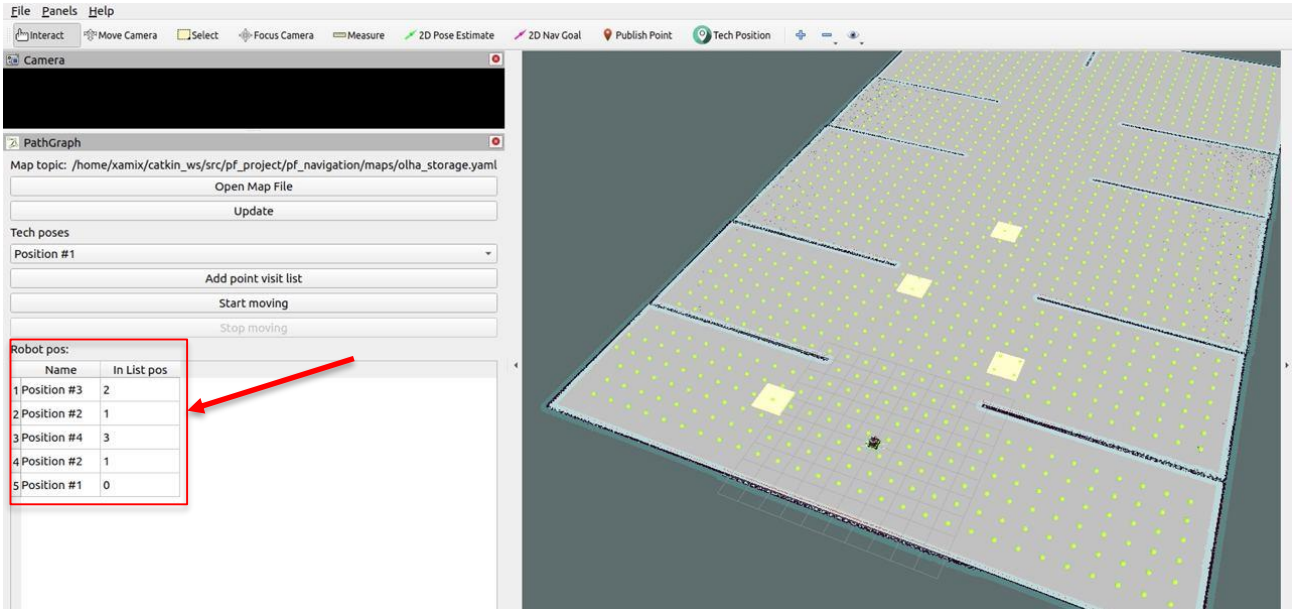


Рисунок 3.11 — Зображення доданих технологічних позицій

Далі визначається технологічна позиція за допомогою Tech Position. При натисненні на кнопку Start moving прокладається шлях від робота до цілі (рис. 3.12).

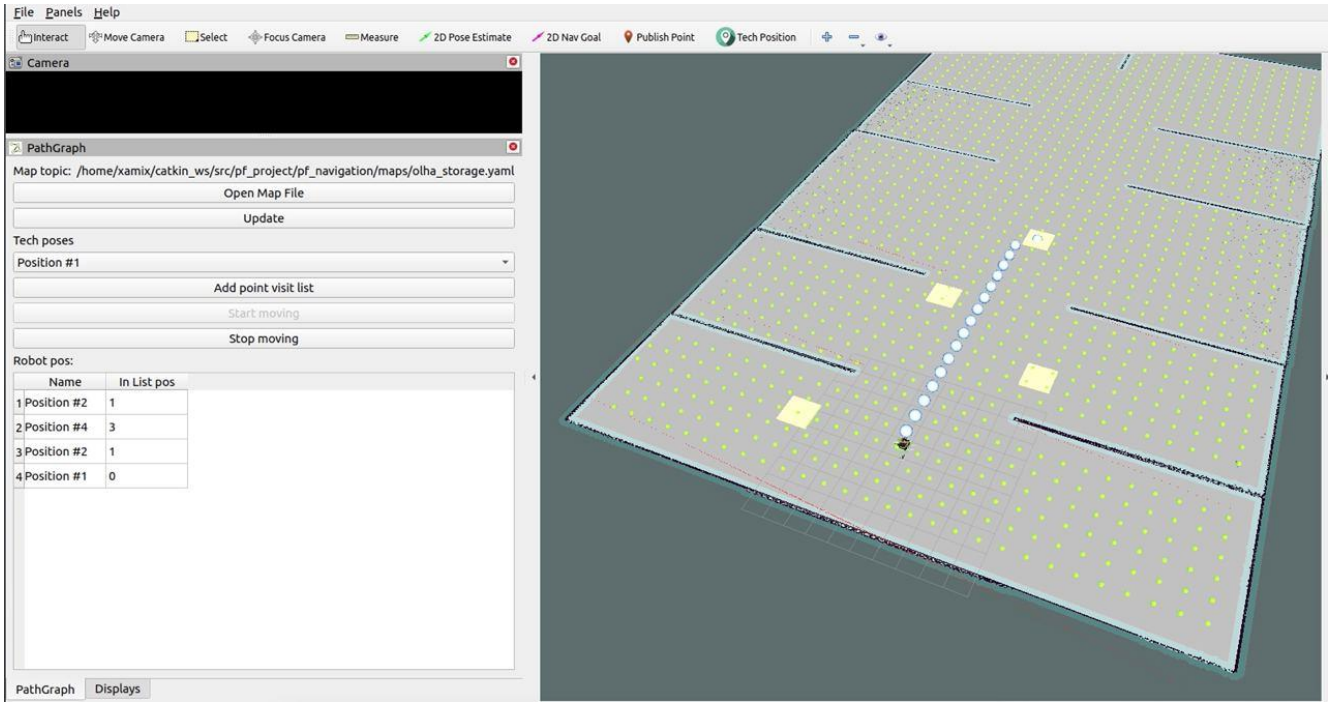


Рисунок 3.12 — Візуалізація шляху у середовищі RVIZ

Екранні форми, які показують процес роботи із плагіном, наведені у Додатку И.

## 3.4 Висновки до розділу

У даному розділі складено та виконано детальний аналіз роботи по створенню плагіна пошуку оптимальних шляхів для мобільного транспортного робота в створеній моделі промислового приміщення. Розроблена підсистема складається з трьох взаємопов’язаних модулів, а саме: модуля обробки мапи прохідності та побудова графу шляхів, модуля задання користувачем технологічних позиції, їх зберігання на мапі, створення списку їх відвідування, модуля для розрахунку оптимального шляху пересування між ними.

Було виконано інтеграцію розробленої підсистеми до ROS, а саме середовища RVIZ. Було реалізовано необхідні для повноцінного функціонування даної підсистеми топіки та відповідні для них ноди, встановлено комунікацію між ними та перевірено справність обміну інформацією. Проведено візуалізацію руху мобільного транспортного робота по оптимальному маршруту в середовищі Gazebo та синхронного відображення маршруту в RVIZ.

# 4 КРИТЕРІЙ ОЦІНКИ ВАРТОСТІ ЕКСПЛУАТАЦІЇ МОБІЛЬНОГО ТРАНСПОРТНОГО РОБОТА

В даному розділі буде розглянутий критерій оцінки вартості володіння, підтримки та використання розробленої системи для керування роботами в умовах промислових приміщень з виконанням типових технологічних операцій. Для цього обрано один з таких методів, часто використовуваних в сучасних економічних розрахунках вартості автоматизованих та комп'ютеризованих систем, - ТСО (Сукупна вартість володіння). Це сучасний метод оцінки ефективності придбання, володіння та експлуатації обладнання та його комплектуючих.

Передумовами для вивчення цього питання стало різке збільшення попиту у сфері автоматизації виробництва. За останніми даними Міжнародної федерації робототехніки (IFR), використання роботів на виробництві прискорюється в усьому світі й у середньому в промисловості на 10 тисяч працівників припадають 74 роботи. Щільність роботів в Європі на виробництві найвища й становить 99 одиниць, в Америці це 84, а в Азії – 63 одиниці. Десятку найбільш автоматизованих країн у світі очолює Південна Корея, Сінгапур, Німеччина, Японія, Швеція, Данія, США, Італія, Бельгія та Тайвань. Ідеться не про розробку інновацій у робототехніці, а саме про темпи впровадження роботів у промисловості.

Україна не увійшла до переліку 40 країн, що використовують роботів у виробництві, але роботизація виробничих процесів визначена як складова державної політики у сфері інновацій на найближчі роки [33].

Більше мобільних транспортних роботів з меншими експлуатаційними витратами – сучасна мета для підприємств, які поставили завдання запуску висококонкурентних продуктів і послуг в найкоротші терміни і з мінімальними витратами.

Наприклад, за статистикою [34], оплата праці складського персоналу складає близько 65% від загальних операційних витрат. До того ж така робота потребує великої швидкості та значних фізичних навантажень. Деякі склади здатні обробляти більше 10 000 одиниць за день, а в середньому працівники проходять від

12 до 20 кілометрів за одну зміну. В таких умовах працівнику фізично важко швидко та правильно виконувати завдання. Вірогідність помилок, пов’язаних із людським фактором, зростає. Щоб збільшити продуктивність складських приміщень та знизити ризики помилок, середні та малі підприємства розпочали інвестувати у автоматизацію складських процесів. Цей напрямок став активно розвиватися в Україні у 2020 році.

Промислові мобільні роботи перестали бути привілеєю важкої промисловості або великих заводів, доля яких у загальній кількості діючих виробництв складає лише 0.5%. Інші діючі підприємства – це виробництва малого та середнього бізнесу в Україні, як свідчать дані наведені у таблиці 4.1, що є офіційною статистичною інформацією Державної служби статистики України [35] – їх доля складає 99.5% від загальної кількості. Це надзвичайно великий сегмент для впровадження мобільних роботів. Проведені дослідження показали необхідність швидкого адаптування у застосуванні мобільних транспортних роботів в умовах малого та середнього бізнесу.

Таблиця 4.1 – Кількість промислових підприємств в Україні (за обсягом виробництва)

| № | Роки | Загальна кількість, одиниць | Великі підприємства | | Середні та малі підприємства | |
|---|---|---|---|---|---|---|
| | | | Кількість, одиниць | Доля у загальній кількості діючих підприємств, % | Кількість, одиниць | Доля у загальній кількості діючих підприємств, % |
| 1 | 2020 | 47806 | 243 | 0.5 | 47563 | 99.5 |
| 2 | 2018 | 44425 | 237 | 0.5 | 44188 | 99.5 |
| 3 | 2016 | 38555 | 208 | 0.5 | 38347 | 99.5 |

## 4.1 Критерій оцінки ефективності системи

Транспортні мобільні роботи (ТМР) довели, що підвищують продуктивність і знижують витрати на автоматизацію складських та промислових приміщень. Хоча витрати на ТМР продовжують знижуватися, навіть якщо їх можливості збільшуються, часто виникають приховані та значні витрати, пов’язані з їх розгортанням, підтримкою, обслуговуванням та модернізацією. Впроваджуючи ТМР на підприємстві, важливо враховувати як початкові, так і поточні зусилля з налаштування, розгортання та реконфігурації, які можуть стати дорогими, якщо роботизованою системою складно користуватися, та якщо вона не є достатньо гнучкою [22].

Налаштування ТМР вимагає створення як мінімум карт, опорних точок і точок виконання операцій. Для багатьох компаній отримання робота з коробки та налаштування його в автономному режимі є складним процесом. Це може призвести до прихованих витрат на впровадження, які можуть включати:

- втрачений час на перебудову або адаптацію будівлі;
- вартість витраченого часу постачальника або робототехніка для налаштування системи;
- витрачений час на навчання співробітників підприємства;
- години, витрачені на створення та редагування карт приміщень;
- загальна довжина шляху, який проходить робот;
- кількість актів запуску та зупинки двигунів робота (або електроприводів), що впливає на їх ресурс;
- кількість вимушених поворотів робота, що впливає на зношення шасі.

Конфігурація робота є частиною процесу впровадження автоматизованої системи, але залежність від мережі, інтеграція з іншими фізичними та програмними системами, а також прийняття робочою силою також є критичними факторами.

Якщо система не є інтуїтивно зрозумілою і зручною для кінцевого користувача, вони будуть розчаровані, оскільки витрачатимуть велику кількість часу, з’ясовуючи причини несправностей або намагаючись отримати допомогу від

постачальника автоматизованої системи. Ресурс апаратних частин МР також не є вічним, й правильне їх використання значно зменшує витрати на ремонт та інше техобслуговування [21].

Для оцінки економічної ефективності розробленого алгоритму пошуку найкоротшого шляху для ТМР та переваг його використання буде застосований метод ТСО. На сьогоднішній день ТСО не використовувався для розрахунку оптимальності функціонування системи управління рухом МР, тому у пункті 4.3 буде запропоновано критерії для оцінки вартості руху та використання систем на базі МР в промислових приміщеннях.

## 4.2 Опис методу ТСО

ТСО - це сумарна ціна придбання активу плюс витрати на експлуатацію. Оцінка загальної вартості володіння означає більш детальну картину того, що являє собою продукт і яка його вартість з часом. ТСО – це сума всіх очевидних і всіх прихованих витрат, пов’язаних з активом за загальний період володіння, за загальний термін корисного використання та/або протягом терміну експлуатації активу. Обираючи одну із альтернатив у рішенні про придбання тієї чи іншої системи, потенційні клієнти повинні дивитися не тільки на короткострокову ціну товару, відому як його купівельна ціна, але й на його довгострокову ціну, яка є його повною вартістю володіння. Це довгострокові витрати та витрати, понесені протягом строку корисного використання продукту та остаточної утилізації. Предмет з нижчою загальною вартістю володіння є кращим в довгостроковій перспективі [23].

ТСО є важливим інструментом підтримки стратегічного управління витратами. Це складний підхід, який вимагає від компанії визначити, які витрати вона вважає найбільш релевантними або значущими при придбанні, володінні, використанні та подальшому розпорядженні. На додаток до ціни, сплаченої за товар, ТСО може включати витрати, понесені при закупівлі для розміщення

замовлень, дослідження та кваліфікації постачальників, транспортування, отримання, перевірки, бракування, зберігання та утилізації [23].

Одним із видів аналізу ТСО є підтримка рішення щодо вибору та оцінки компанії. Традиційні підходи включають вибір і утримання компанії на основі лише ціни або якісної оцінки діяльності з використанням категоричних або зважених точкових/матричних підходів. Хоча останнім надається перевага, ніж орієнтація на «тільки ціну», такі підходи, як правило, не акцентують увагу на витратах, пов'язаних з усіма аспектами діяльності постачальника, і загалом ігнорують внутрішні витрати. Перевірка таких витрат є сильною стороною підходу ТСО. ТСО застосовується практично до кожного типу закупівель і включає витрати на закупівлю, пов'язані з конкретним постачальником.

Загальна вартість володіння враховується компаніями та фізичними особами, коли вони хочуть купити активи та зробити інвестиції в капітальні проекти. Для бізнесу вартість придбання та витрати на експлуатацію та обслуговування часто вказуються окремо у фінансовій звітності. Перші обліковуються як капітальні витрати, а другі є частиною операційних витрат. Комплексний аналіз вартості володіння є звичайною практикою для бізнесу [25].

Компанії використовують загальну вартість володіння в довгостроковій перспективі як основу для аналізу ділових угод. Розгляд загальної вартості володіння – це спосіб більш цілісного підходу, який оцінює покупку з широкої точки зору. Цей аналіз включає початкову ціну покупки, а також усі прямі та непрямі витрати. Хоча прямі витрати можна легко описати, компанії найчастіше намагаються проаналізувати всі потенційні непрямі витрати, які можуть мати значний вплив при прийнятті рішення про завершення покупки або проекту.

Принцип ТСО полягає в тому, щоб знайти всі приховані витрати на купівлю товару, ціна товару становить лише 25,0 % від загальної вартості використання продукту. Для кожного продукту вони дуже індивідуальні, тому використання аналізу ТСО є дуже вичерпним. На відміну від традиційних методів економії витрат, які зосереджені на всіх сферах, аналіз ТСО підтримує стратегічне

управління коштами, що означає, що весь аналіз враховує вплив рішень про закупівлю в організації [24].

Таким чином, ТСО забезпечує вартість продукту з точки зору клієнта. Він пропонує чіткий огляд усіх витрат, які несе система, доки вона утримується клієнтами. Прикладами витрат, включених до розрахунків ТСО, є витрати на придбання, навчання персоналу, витрати на електроенергію, технічне обслуговування та утилізацію, а також витрати на закінчення терміну експлуатації.

Основні розрахунки ТСО проводяться за загальною формулою (5) [24]:

$$TCO = \Sigma C_{ownership} + \Sigma C_{service} + \Sigma C_{idleTime} \quad (5)$$

де $C_{ownership}$ - це витрати на закупівлю;

$C_{service}$ - витрати на обслуговування;

$C_{idleTime}$ - витрати на простій.

## 4.3 Застосування ТСО для оцінки руху мобільного робота

Для детального розрахунку ТСО необхідно проаналізувати прямі та непрямі витрати [26], які можуть виникнути при використанні розроблюваної підсистеми формування руху робота (Таблиця 4.2).

Таблиця 4.2 – Прямі та непрямі витрати

| | Прямі витрати | Непрямі витрати |
|---|---|---|
| Витрати на закупівлю | - початкова ціна за підсистему пошуку шляху (26000 грн);<br>- придбання транспортного мобільного робота (80000 грн);<br>- налаштування системи (10000 грн); | - навчання працівників (5000 грн); |
| Витрати на обслуговування | - заміна комплектуючих робота (2000 грн/місяць);<br>- оновлення підсистеми (2000 грн); | - витрати на електроенергію (5000 грн/місяць);<br>- витрати на Інтернет (2000 грн/місяць); |

| | | |
|---|---|---|
| | | - переналаштування мапи (5000 грн); |
| Витрати на простій | - запланований простій через оновлення підсистеми (10000 грн). | - аварійна зупинка робота або його частин (50000 грн);<br>- збій системи (50000 грн). |

Одним з головних показників ефективності роботи даної підсистеми пошуку шляху є довжина шляху [21]. Вона безпосередньо впливає на швидкість виконання операцій роботом, витрати на електроенергію, знос деталей, бо чим менше проїде робот, тим повільніше зносяться його комплектуючі, і т.д. Тому для підсистеми пошуку шляху вартість одного метра шляху буде розрахований як відношення довжини шляху до вартості експлуатації за формулою (6):

$$TCO_{month} = \frac{avgPath * pathPerMonth}{\Sigma C_{initialCosts} + \Sigma C_{hiddenCosts}} \quad (6)$$

де *avgPath* – середня довжина шляху відвідування технологічних позицій, обчислена за формулою (8);

*pathPerMonth* – довжина шляху, яку пройшов робот за місяць, розраховане за формулою (7);

$C_{initialCosts}$ – це прямі витрати, $C_{hiddenCosts}$ – непрямі витрати.

$$pathPerMonth = avgPath * repeatCount * 28 \quad (7)$$

де *repeatCount* – кількість повних об'їздів роботом 4 технологічних позицій за день.

$$avgPath = \frac{\sum_{i=0}^{n} pathLength}{n} \quad (8)$$

де *pathLength* – середня довжина шляху, виміряна на основі *n* вимірів.

Для зручності спочатку окремо розрахуємо вартість експлуатації як суму прямих та прихованих коштів.

$$\Sigma C_{initialCosts} = 130000 \text{ грн}$$

$$\Sigma C_{hiddenCosts} = 117000 \text{ грн}$$

За замірами роботи підсистеми, проведеними з застосування модифікованого алгоритму A* повний шлях для маршруту з відвідуванням 4 технологічних позицій в середньому складає 1688.12 м для приміщення розміром 50 на 50 метрів, що при виконанні об'їзду цих точок за день близько 100 разів буде складати 3084060 м за місяць. Тоді значення ТСО за один місяць експлуатації складатиме:

$$pathPerMonth = 1688.12 * 100 * 28 = 4726736 \text{ м}$$

$$TCO_{month} = \frac{1688.12 * 4726736}{247000} = 32304.85 \text{ грн}$$

Отримано, що критерій оцінки ТСО за місяць складає 24.85% для обраного прикладу.

## 4.4 Порівняння критерію ТСО зі значеннями роботи інших алгоритмів

При розробці підсистеми пошуку шляху було використано модифікацію функції оцінки оптимальності шляху, основаної на Евклідовій формулі. Для аналізу ефективності роботи системи на цій модифікації ТСО необхідно порівняти зі значеннями ТСО, розрахованими з використанням інших функцій оцінки. Для дослідження обрано функції Чебишева, Octile та Manhattan (рис. 4.1).

Для алгоритму A* з різними модифікаціями проведено заміри довжини шляху та кількості поворотів за однакових умов. Проведено розрахунки ТСО та представлено отримані результати у Таблиці 4.3 та на рис. 4.2.

Таблиця 4.3 – Заміри довжини шляху для різних модифікацій

| Модифікація | Середня довжина шляху (4 технологічні позиції), м | Кількість поворотів | ТСО,% |
|---|---|---|---|
| Manhattan | 1859.1 | 41 | 30.1% |
| Euclide | 1688.12 | 22 | 24.85% |
| Octile | 1724.0 | 25 | 25.92% |
| Chebyshev | 2265.29 | 45 | 44.39% |

За отриманими даними, можна помітити, що значення ТСО для алгоритму A* з модифікацією функції оцінки оптимальності шляху, основаної на Евклідовій формулі менше, ніж на інших його модифікаціях.

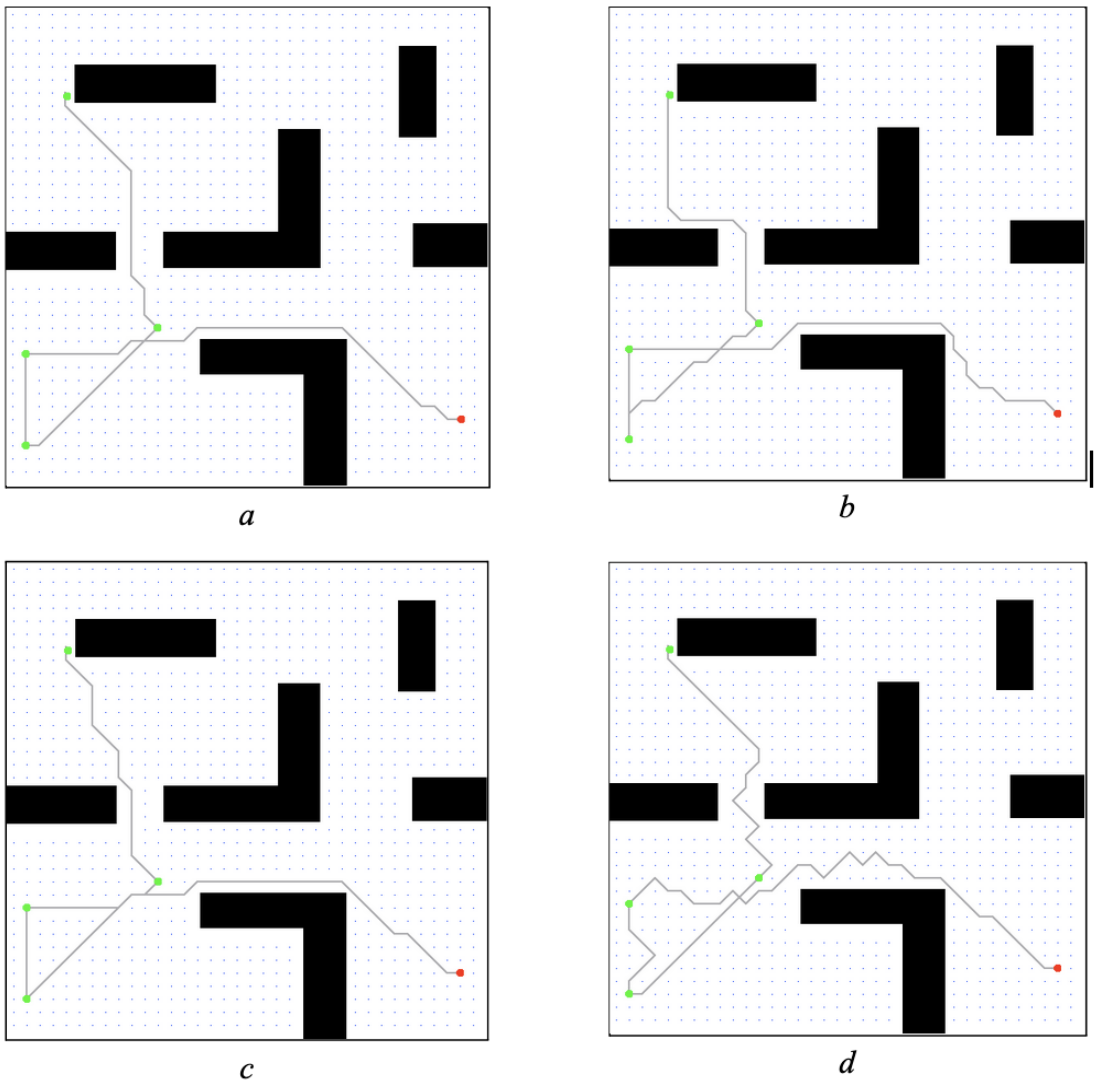


Рисунок 4.1 — Візуалізація роботи підсистеми пошуку шляху з різними функціями оцінки: *a* - функція Euclide, *b* - функція Manhattan, *c* - функція Octile, *d* - функція Чебишева

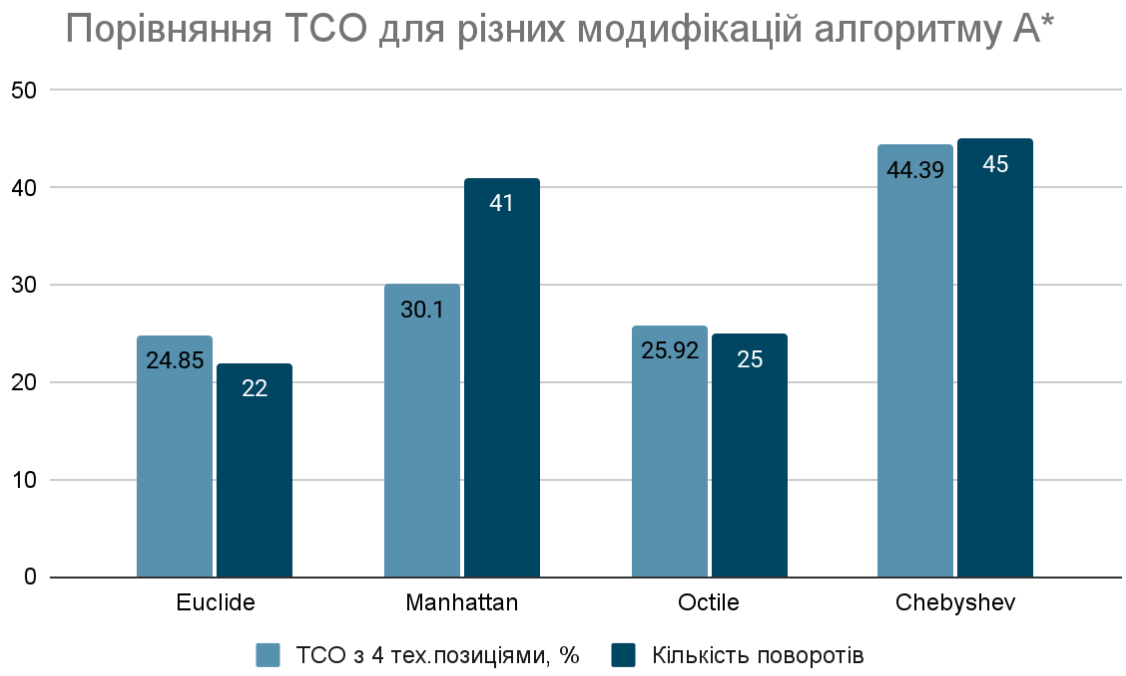


Рисунок 4.2 — Порівняння ТСО для різних модифікацій алгоритмів

## 4.5 Висновки до розділу

В четвертому розділі було проаналізовано актуального даного дослідження з економічної точки зору та запропоновано критерій для оцінки вартості експлуатації на основі критерію ТСО. ТСО є популярним методом оцінки ефективності придбання, володіння та експлуатації обладнання та його комплектуючих та мало використовувався у робототехніці. За допомогою запропонованого критерію розраховано показник ТСО для модифікації алгоритму A*, використовуваного в розробленій підсистемі пошуку шляху. Виконано порівняння цього значення зі значеннями ТСО для інших модифікацій на базі різних функцій оцінки.

# 5 МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

## 5.1 Опис ідеї проекту

Таблиця 5.1 – Опис ідеї стартап-проекту

| Зміст ідеї | Напрямки застосування | Вигоди для користувача |
|---|---|---|
| Проектування та розробка системи моделювання маршруту мобільного робота, що дозволить ефективно здійснювати перевезення вантажу в промислових приміщеннях оптимальним чином по місцях виконання типових технологічних операцій. | Забезпечення керування роботизованими системами складських приміщень та інших промислових об’єктів. | Зменшення витрат на програмне забезпечення, через те, що дана система може працювати з приміщеннями різного розміру та конфігурації, та самостійно враховує габарити рухомого робота, для прокладання маршрутів руху. Ефективне використання ресурсів часу та пального/електроенергії рухомими вантажними роботами. |
| | Створення автоматизованої системи управління роботизованим устаткування для переміщення вантажу з мінімізацією витрат пального/електроенергії шляхом переміщення оптимальним чином. | |

Таблиця 5.2 – Опис ідеї стартап-проєкту

| № | Техніко-економічні характеристики ідеї | Продукція конкурентів | | | | W (слабка сторона) | N (нейтральна сторона) | S (сильна сторона) |
|---|---|---|---|---|---|---|---|---|
| | | Мій проєкт | Haі Robotics | MECA LUX | FORSTOR | | | |
| 1 | Можливість роботи в довільному | Так, карта будується на основі показників датчиків | Ні | Ні | Ні | | + | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | приміщенні | | | | | | | |
| 2 | Потреба в спеціалізованих стелажах | Ні | Так | Так | Так | | | + |
| 3 | Готовність до змін планування приміщення | Потрібно лише перегенерувати карту приміщення | Потребує встановлення додаткових стелажів тощо | Потребує встановлення додаткових стелажів тощо | Потребує встановлення додаткових стелажів тощо | | | + |
| 4 | Витрати за користування | За користування | За наданий план/проєкт | За наданий план/проєкт | За наданий план/проєкт | | | + |

5.2 Технологічний аудит ідеї проєкту

Таблиця 5.3 – Технологічна здійсненність ідеї проєкту

| № | Ідея проєкту | Технології її реалізації | Наявність технологій | Доступність технологій |
|---|---|---|---|---|
| 1 | Створення підсистеми розрахунку маршрутів руху роботів по промислових приміщеннях з | Велика кількість алгоритмів розбиття карти прохідності на граф маршрутів та велика кількість алгоритмів | Алгоритми побудови графів доріг: граф видимості, граф маршрутних точок, сітка. Алгоритми побудови шляху на графах: A*, | Опис алгоритмів знаходяться в вільному доступі, деякі модифікації в наукових статтях. Відсутні особливі вимоги |

| | | | | |
|---|---|---|---|---|
| | використання м алгоритму пошуку шляхів та графу маршрутів. | пошуку шляху на графах. | алгоритм Дейкстри, DFS. | до роботів та приміщення. |
| 2 | Створення підсистеми розрахунку маршрутів руху роботів по промислових приміщеннях с використання м даних з датчиків. | Класичний варіант руху, спираючись на показники дальномірів та відносної позиції робота та цілі. | Алгоритми класу Bug. | Опис алгоритмів знаходяться в вільному доступі. Відсутні особливі вимоги до роботів та приміщення, але високі вимоги до якості датчиків робота. |
| 3 | Створення підсистеми розрахунку маршрутів руху роботів по наперед закріпленими рейками для пересування роботів. | Класичний варіант з наперед сконструйованими дорогами (коліями, доріжками тощо) для руху робота. | Будь-які алгоритми побудови шляху на графах. | Опис алгоритмів знаходяться в вільному доступі. Особливі вимоги до приміщення (повинна бути встановлення інфраструктура для переміщення робота). |
| *Обрана технологія реалізації ідеї проекту:1* | | | | |

Висновок: технологічна реалізація продукту – можлива, вибрана технологія №1

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 5.4 – Попередня характеристика потенційного ринку

| № | Показники стану ринку | Характеристика |
|---|---|---|
| 1 | Кількість головних гравців, од | 3 |
| 2 | Загальний обсяг продаж, грн./ум.од | 1 500 000 грн/ум. од. за місяць |
| 3 | Динаміка ринку | Зростає |
| 4 | Наявність обмежень для входу | Необхідність високонадійної, енергоефективної та відмовостійкої системи |
| 5 | Специфічні вимоги до стандартизації та сертифікації | ISO |
| 6 | Середня норма рентабельності в галузі або по ринку, % | 65% |

Висновок: враховуючи кількість головних гравців по ринку, зростаючу динаміку ринку, невелику кількість конкурентів та середню норму рентабельності можна зробити висновок, що на даний момент, ринок для входження стартап-продукту є привабливим.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

| № | Потреба, що формує ринок | Цільова аудиторія | Відмінності у поведінці цільових груп клієнтів | Вимоги споживачів до товару |
|---|---|---|---|---|
| 1 | Потреба в системі керування великими | Власники складів, точок оптової продажі, сільськогоспода | Ступінь автоматизованості приміщення | Проста масштабованість, відсутність специфічних вимог до |

| | | | | |
|---|---|---|---|---|
| | складськими приміщеннями | рські підприємства | | промислового об'єкту, здатність запустити проект за короткий термін, зручні умови оплати |
| 2 | Забезпечення автоматизації промислових приміщень з динамічною топологією | Промислові підприємства, які у разі зміни планування складу, повинні заново налаштовувати логістику на них | Площа промислової зони, топологічна складність промислової зони | |

Таблиця 5.6 – Фактори загроз

| № | Фактор | Зміст загрози | Можлива реакція компанії |
|---|---|---|---|
| 1 | Відсутність можливості побудови карти приміщення, або приміщення динамічно змінюється. | Система спирається на карту приміщення та фізичні розміри керованого робота, її побудова є критичною для функціонування системи. | Мануальна побудова карти з урахуванням фізичних розмірів об'єктів, або її автоматичне оновлення у разі суттєвих змін. |
| 2 | Відтік кадрів | Розробка та доопрацювання системи, а також розробка власних (за необхідністю) роботів потребує висококваліфікованих працівників | Забезпечення сприятливих умов праці, заохочувальні акції та соц. пакети, оплачувані відпустки та гнучкий графік |

Таблиця 5.7 – Фактори можливостей

| № | Фактор | Зміст можливості | Можлива реакція компанії |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | Зростання попиту в автоматизованих приміщеннях | Внаслідок збільшення навантаженості на складські та інші промислові приміщення виникає необхідність в їх автоматизації | Переговори з власниками складів та інших промислових приміщень про інвестиції та можливі варіанти партнерства |
| 2 | Зростання кваліфікації наявних працівників | Внаслідок доопрацювання та вдосконалення наявної системи, рівень знань працівників збільшиться, що дасть можливість значно покращити систему | Можлива розробка нового функціоналу та обслуговування складніших промислових приміщень |
| 3 | Збільшення навантаження на складські приміщення | Внаслідок зростання навантаження на існуючі складські приміщення, все більше власників будуть намагатись їх автоматизувати | Переговори з власниками складів та інших промислових приміщень про інвестиції та можливі варіанти партнерства |

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

| № | Особливості конкурентного середовища | В чому проявляється дана характеристика | Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною) |
|---|---|---|---|
| 1 | Тип конкуренції: монополістична | Відсутність повних аналогів на ринку | Надання унікальних послуг та сервісів для задоволення потреб користувача |

| | | | |
|---|---|---|---|
| 2 | Рівень конкурентної боротьби: світовий | Продукт доступний у всіх країнах світу. | Надання послуг усім зацікавленим підприємствам та компаніям. |
| 3 | Галузева ознака: внутрішньогалузева | В межах галузі робототехнічних та інформаційних технологій. | Створення зрозумілої документації, підтримка і супровід впровадження проекту. |
| 4 | Конкуренція за видами товарів: товарно-видова | Конкуренція в межах одного виду товарів. | Забезпечення функціоналу, відсутнього у компаній-конкурентів, та надання супроводження та підтримку клієнтів. |
| 5 | Характер конкурентних переваг: цінова та не цінова | Орієнтація на варіативність функцій товару. | Формування набору функціоналу та особливостей в залежності від потреб клієнта. |
| 6 | За інтенсивністю: марочна | Створення власного товарного знаку. | Забезпечення впізнання бренду серед інших компаній. |

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

| Складові аналізу | Прямі конкуренти в галузі | Потенційні конкуренти | Постачальники | Клієнти | Товари-замінники |
|---|---|---|---|---|---|
| | Haі Robotics<br>MEC ALUX<br>FORSTOR | Доступ до ресурсів та каналів збуту | Конкуренція постачальників | Система інформації та контроль якості | Лояльність споживачів |
| Висновки | Мала | Так, на ринок можна | Розширення клієнтської бази формує | Клієнти хочуть отримати | Споживачі можуть бути не готові |

| | | | | | |
|---|---|---|---|---|---|
| | | виходити. Можлива поява конкуренті в враховуюч и динаміку розвитку робототехн ічної сфери. | певний монополізм на ринку, тому необхідно буде вести конкурентну боротьбу. | максималь но оптимальн е рішення за співвідношенням ціни та функціона лу. | довіряти новій системі. |

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з наявних продуктів має своє специфічне рішення з певними вимогами до роботів та приміщень, свої позитивні та негативні сторони щодо вирішення певних типів задач, то робота та вихід на даний ринок є можливою і реалізованою задачею тому, що нове рішення буде орієнтоване на простоту та відсутність спеціальних вимог до приміщень.

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

| № | Фактор конкурентоспроможності | Обґрунтування |
|---|---|---|
| 1 | Актуальність технології | Створений програмний комплекс використовує сучасні алгоритми та методики, що спрощує його підтримку та розвиток |
| 2 | Простота розвертання | Система не потребує особливих умов для встановлення та функціонування |
| 3 | Простота масштабування | Система може самостійно масштабуватися за необхідністю |

| | | |
|---|---|---|
| 4 | Зручна модель оплати | Оплата здійснюється за встановлення системи та її підтримки, або переналаштування |
| 5 | Універсальність системи | Система може функціонувати на будь-яких технологічних об'єктах, та з будь-якими мобільними роботами |

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін системи

| № | Фактор конкурентоспроможності | Бали 1-20 | Рейтинг товарів-конкурентів у порівнянні з запропонованим | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | -3 | -2 | -1 | 0 | +1 | +2 | +3 |
| 1 | Актуальність технології | 18 | 3 | 1 | 2 | | | | |
| 2 | Простота розвертання | 19 | 2 | 1 | 3 | | | | |
| 3 | Простота масштабування | 19 | | 2 | 1 | 3 | | | |
| 4 | Зручна модель оплати | 16 | | 2 | 3 | 1 | | | |
| 5 | Універсальність | 18 | 3 | 2 | 1 | | | | |

При проведенні аналізу порівняно три фірми-конкурента: 1 - Hai Robotics, 2 - MECALUX, 3 - FORSTOR. За результатами можна зробити висновок, що дані обраних компаній мають нижчий рейтинг в обраному сегменті ринку відносно запропонованого рішення через наявність додаткових вимог до приміщення та обладнання.

Таблиця 5.12 – SWOT аналіз стартап-проєкту

| | |
|---|---|
| Сильні сторони (S):<br>- Відсутність вимог до технологічного об'єкту<br>- Простота масштабування<br>- Відсутність вимог до вантажних роботів<br>- Простота розвертання | Слабкі сторони (W):<br>- Відсутність значного об'єму даних про тестування системи на практиці<br>- Відсутність актуальності для малих компаній, та великий поріг входу в довіру до великих компаній |
| Можливості (O):<br>- Використання нових підходів та оптимізація | Загрози (T):<br>- Зменшення попиту на автоматизовані робототехнічні системи<br>- Зменшення кількості висококваліфікованих працівників |

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

| № | Альтернатива (орієнтовний комплекс заходів) ринкової поведінки | Ймовірність отримання ресурсів | Строки реалізації |
|---|---|---|---|
| 1 | Надання безкоштовного терміну сервісного обслуговування, для апробації рішення | Висока, тому, що відсутні прямі аналоги | 2-3 місяці |
| 2 | Залучення реклами для підприємств та контактів в сфері робототехнічних систем | Залучення власних коштів для реклами системи | 2-3 місяців |

| | | | |
|---|---|---|---|
| 3 | Переговори з власниками малих та середніх за розміром технологічних підприємств | Можливість використання ресурсу в рамках партнерства та розвитку | 6-12 місяців |

5.4 Розроблення ринкової стратегії проекту

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

| № | Опис профілю цільової групи потенційних клієнтів | Готовність споживачів сприйняти продукт | Орієнтовний попит в межах цільової групи (сегменту) | Інтенсивність конкуренції в сегменті | Простота входу у сегмент |
|---|---|---|---|---|---|
| 1 | Середні технологічні підприємства | Висока зацікавленість в розвитку та спрощення логістики за помірну ціну | Високий<br>Ефективне рішення враховуючи розвиток підприємств | Мала<br>Аналоги для середніх підприємств за помірну ціну відсутні | Просто<br>Доступна значна ніша |
| 2 | Великі технологічні підприємства | Низька зацікавленість в нових та сторонніх рішеннях | Низький<br>Можливість використовувати складні та комплексні рішення, а також власні розробки | Середня<br>Частка великих підприємств мають свої рішення або використовують наявні на ринку | Складно<br>Закрита корпоративна ніша<br>Складно переоснастити вже робочі автоматизовані |

| | | | | | |
|---|---|---|---|---|---|
| | | | | складні системи | ділянки та приміщення |
| Які цільові групи обрано: 1 | | | | | |

Відповідно до проведеного аналізу можна зробити висновок, що підходящою цільовою групою для розповсюдження даного програмного продукту є середні технологічні підприємства, так як запропоноване рішення є простим, зрозумілим, доступним за ціною та легко впроваджуваним.

Таблиця 5.15 – Визначення базової стратегії розвитку

| Обрана альтернатива розвитку проєкту | Стратегія охоплення ринку | Ключові конкурентоспроможні позиції відповідно до обраної альтернативи | Базова стратегія розвитку |
|---|---|---|---|
| Забезпечення оптимальності, простоти та дешевизни рішення | Акцент на простоті масштабування та розвертання, відсутності спеціальних вимог до приміщень та обладнання | Нова система керування роботизованими складами та технологічними ділянками яка не має прямих аналогів | Стратегія спеціалізації |

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

| Чи є проєкт «першопрохідцем» на ринку | Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів? | Чи буде компанія копіювати основні характеристики товару конкурента, які? | Стратегія конкурентної поведінки |
|---|---|---|---|
| Враховуючи актуальність технологій та застосований підхід, так є першопрохідцем | Так, буде шукати нових клієнтів, що потребують сучасного та простого рішення | Ні, не буде | Стратегія зайняття вільної ніші |

Таблиця 5.17 – Визначення стратегії позиціонування

| № | Вимоги до товару цільової аудиторії | Базова стратегія розвитку | Ключові конкурентоспроможні позиції власного стартап-проєкту | Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту |
|---|---|---|---|---|
| 1 | Простота, масштабованість, відсутність складних вимог до підприємства, дешивизна | Стратегія диференціації | Система не має специфічних вимог до підприємства, легко масштабується, здатна працювати майже з будь-яким | Простота, легкість в провадженні та підтримці, дешивизна |

| | | | | |
|---|---|---|---|---|
| | | | обладнанням та є досить дешевою | |

Відповідно до проведеного аналізу можна зробити висновок, що стартап-компанія вибирає як базову стратегію розвитку – стратегію диференціації, а як стратегію конкурентної поведінки - стратегію зайняття вільної ніші.

## 5.5 Розроблення маркетингової програми стартап-проекту

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

| № | Потреба | Вигода, яку пропонує товар | Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити) |
|---|---|---|---|
| 1 | Автоматизація перевезення вантажу на складському приміщенні | Система автоматично складає маршрути руху вантажного робота | Система може працювати в приміщенні довільної форми |
| 2 | Пришвидшення виконання типових операцій (завантаження та розвантаження) | Система будує оптимальний маршрут руху на основі алгоритму A* | Це достатньо оптимальний алгоритм для шляху й мінімізації палива та електроенергії |

| | | | |
|---|---|---|---|
| 3 | Усунення фактору людської помилки та усунення травмувань | Система не потребує присутності людей на склад | Потрібен лише один оператор для початкових налаштувань системи |

Таблиця 5.19 – Опис трьох рівнів моделі товару

| Рівні товару | Сутність та складові | | |
|---|---|---|---|
| 1. Товар за задумом | Підсистема моделювання маршрутів руху мобільного транспортного робота | | |
| 2. Товар у реальному виконанні | Властивості/характеристики | М/Нм[2] | Вр/Тх/Тл/Е/Ор[3] |
| | Простота використання | М | Тл |
| | Дешева собівартість | Нм | Вр |
| | Масштабованість | М | Тх |
| | Швидкість роботи | Нм | Тх |
| | Універсальність | М | Тх |
| | Якість: ISO 19649:2017 | | |

[2] *М/Нм – монотонні/немонотонні*
[3] *Вр/Тх/Тл/Е/Ор – вартісні/ технічні/ технологічні/ ергономічні/ органолептичні*

| | |
|---|---|
| | Спосіб взаємодії: плагін для пошуку оптимального шляху та інтерфейс RVIZ для створення карти приміщення через API ROS |
| | Марка: Transport_Rob |
| 3. Товар із підкріпленням | До продажу: тестова модель системи, наявність документації. |
| | Після продажу: супровід та підтримка з боку розробників. |
| Товар буде захищено від копіювання патентом на корисну модель. | |

Таблиця 5.20 – Визначення меж встановлення ціни

| Рівень цін на товари-замінники | Рівень цін на товари-аналоги | Рівень доходів цільової групи споживачів | Верхня та нижня межі встановлення ціни на товар/послугу |
|---|---|---|---|
| 300000 - 1500000 грн залежно від тарифу | 300000 - 1500000 грн залежно від тарифу | > 1000000 грн | Від 0 до 1500000 грн залежно плану. |

Таблиця 5.21 – Формування системи збуту

| Специфіка закупівельної поведінки цільових клієнтів | Функції збуту, які має виконувати постачальник товару | Глибина каналу збуту | Оптимальна система збуту |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Задоволення потреб сегменту автоматизації промислових приміщень. | Налагодження зв’язків з можливими замовниками. | Комунікація безпосередньо з власниками складів, теплиць та інших промислових приміщень та пропонування послуг з їх автоматизації. | Власна система збуту |

Таблиця 5.22 – Концепція маркетингових комунікацій

| № | Специфіка поведінки цільових клієнтів | Канали комунікацій, якими користуються цільові клієнти | Ключові позиції, обрані для позиціонування | Завдання рекламного повідомлення | Концепція рекламного звернення |
|---|---|---|---|---|---|
| 1 | | Електронна пошта, телефонний зв’язок, соціальні мережі, бізнес зустрічі | Залучення великої клієнтської бази, можливість доопрацювання та покращення системи | Стимул користувачів до автоматизації їх промислових зон | Система є простою в інтеграції та підтримці, а також має помірну ціну. |

Як результат було створено ринкову (маркетингову) програму, що містить в собі визначення переваг та недоліків в порівнянні з конкурентами, продумано цінову модель та можливі маркетингові зв’язки.

## 5.6 Висновки до розділу

Багато середніх та малих підприємств на поточний час шукають вектори розвитку та зменшення витрат на утримання промислових приміщень одним з таких векторів виступає автоматизація та роботизація підприємства, тому зростає попит на автоматизовані складські та промислові рішення для перевезення вантажу тощо. Враховуючи те що поточні гравці ринку пропонують рішення які вимагають певного переобладнання наявних приміщень у клієнтів, це не завжди є зручним для них, тому запропоноване у даній магістерській роботі може спростити інтеграцію в наявні підприємства не вимагаючи від них особливих умов. Єдиним бар'єром для входу в ринок може бути не готовність підприємств прийняти новий продукт.

Подальша розробка та доопрацювання проекту може дозволити розширення клієнтської бази, а також різноманітність функціоналу розробленого рішення, що може створити нові партнерські та спонсорські відносини.

В п’ятому розділі було описано та проаналізовано можливості та доцільність створення стартап-проекту на базі розробленої в даній магістерській роботі програмної системи. Було проаналізовано наявних конкурентів на ринку, його обсяг та можливі канали взаємодії та збуту. Встановлено, що створення стартапу є можливим та доцільним, прямої конкуренції в поточний момент немає, а наявні аналоги не мають запропонованих переваг, таких як простота розвертання, ціна та інше.

# ВИСНОВКИ

У даній роботі було розроблено підсистему для формування раціонального маршруту переміщення мобільного транспортного робота при виконанні робочих транспортних операцій для умов роботи в промислових приміщеннях, складах та теплицях з урахуванням мапи приміщення та технологічних позицій для виконання типових операцій.

Магістерська дисертація присвячена вирішенню проблеми планування шляху для мобільних транспортних роботів у промислових приміщеннях шляхом створення підсистеми пошуку оптимального маршруту руху для роботи 4-х колісного мобільного транспортного робота у промислових приміщеннях з наявними технологічними позиціями для виконання типових операцій завантаження, розвантаження та очікування наступної операції.

У ході роботи проаналізовані існуючі рішення роботи транспортних роботів, проте більшість реалізацій направлені на обслуговування великих складів та вимагають великих коштів за спеціальні комплекси та повного переобладнання приміщення спеціальними коліями та кріпленнями. Тому розробка підсистеми пошуку оптимального маршруту для мобільного транспортного робота, якого можна легко адаптувати до умов роботи в невеликих приміщення та за недорогою ціною, є актуальною задачею.

Розглянуто принципи проектування архітектури плагіну, методи алгоритмів розбиття мапи на опорні точки та алгоритмів пошуку найкоротшого шляху. За результатами проведеного аналізу обрано метод сітки для побудови графу шляху та алгоритм A* для пошуку найкоротшого шляху, модифікований використанням функції оцінки Euclide. Побудовано додаток для тестування роботи системи, перед тим, як його застосовувати у плагіні.

Розглянуто взаємодію плагіна та системи ROS та усі основні засоби для взаємодії з роботом. Підключено плагін до середовища RVIZ та візуалізовано отриманий шлях в Gazebo. Проведено тестування роботи усіх частин плагіну: для побудови графу, для визначення технологічних позицій та візуалізації шляху. Дану

підсистему можна інтегрувати у мобільного робота будь-якої конфігурації та застосувати для пошуку шляху у будь-якому приміщенні.

Запропоновано критерій оцінки вартості використання мобільного транспортного робота з розробленої підсистемою пошуку оптимального маршруту на основі методу ТСО з урахуванням критеріїв оптимальності шляху – довжини шляху та кількості поворотів. Проведено порівняльний аналіз роботи алгоритму A* з різними модифікаціями функції оцінки вартості шляху, а саме функції Euclide, функції Чебишева, функції Octile та функції Manhattan. Алгоритм з функцією оцінки Euclide показав найкращий результат - найкоротший шлях та найменшу кількість поворотів.

Також проведено маркетинговий аналіз стартап проекту, де проаналізовано можливості по впровадженню та підтримці даної підсистеми.

Магістерська дисертація була виконана за науково-дослідною роботою № 2000/556 від 10.06.2019р. «Визначення технічних параметрів пошукових дронів для зон катастроф та стихійного лиха».

# СПИСОК ДЖЕРЕЛ


1. Global Path Planning for Unmanned Ground Vehicles / Giesbrecht, J // Technical Memorandum / Giesbrecht, J. – Canada, 2004. – P. 3 – 37.
2. Recent Advances in Robot Path Planning Algorithms: A Review of Theory and Experiment Description / Jahanshahi, Hadi, Najafizadeh Sari, Naeimeh, Pham, Viet-Thanh, Khajepour, Roya, Volos, Christos K. // Nova Science Publishers / Jahanshahi, Hadi, Najafizadeh Sari, Naeimeh, Pham, Viet-Thanh, Khajepour, Roya, Volos, Christos K. – New York, 2020. – P. 13 – 66.
3. Path Planning and Trajectory Planning Algorithms: A General Overview / Gasparetto, Alessandro, Boscariol, Paolo, Lanzutti, Albano, Vidoni, Renato // Mechanisms and Machine Science / Gasparetto, Alessandro, Boscariol, Paolo, Lanzutti, Albano, Vidoni, Renato. – Cham, 2015. – P. 3-27.
4. A fast path planning by path graph optimization / Joo Young Hwang, Jun Song Kim, Sang Seok Lim, Kyu Ho Park // Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, 33(1) / Joo Young Hwang, Jun Song Kim, Sang Seok Lim, Kyu Ho Park. – Canada, 2003. – P. 121–128.
5. Path length comparison in grid maps of planning algorithms: HCTNav, A∗ and Dijkstra / Nafiseh Osati Eraghi, Femando López-Colino, Angel de Castro, Javier Garrido // Design of Circuits and Integrated Systems / Nafiseh Osati Eraghi, Femando López-Colino, Angel de Castro, Javier Garrido. – Madrid, 2014. – P. 1-6.
6. Comparison of optimal path planning algorithms / M. Korkmaz and A. Durdu // 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering / M. Korkmaz and A. Durdu. – Lviv, 2018. – P. 255 – 258.
7. An Overview of Path Planning and Obstacle Avoidance Algorithms in Mobile Robots / Mr Basavanna // International Journal of Engineering and Technical Research / Mr Basavanna. – Gandhinagar, 2019. – P. 478–482.

8. Mobiles Robots – Past Present and Future / Chen, Xiaoqi, Chen, Y.Q., Chase, James // Mobile Robots – State of the Art in Land, Sea, Air, and Collaborative Missions / Chen, Xiaoqi, Chen, Y.Q., Chase, James. – Croatia, 2009 – P. 1–32
9. Research and application of path-finding algorithm based on unity 3D / Zhang He, Minyong Shi, Chunfang Li // 15th International Conference on Computer and Information Science / Zhang He, Minyong Shi, Chunfang Li. – Okayama, 2016. – P. 1–4.
10. Madhav S. Comparison of Efficiency in Pathfinding Algorithms in Game Development / Sanjay Madhav // Game Programming Algorithms and Techniques / Sanjay Madhav. – New York, 2009. – P. 251–264.
11. Wang X. Z. The Comparison of Three Algorithms in Shortest Path Issue / Xiao Zhu Wang // Journal of Physics Conference Series / Xiao Zhu Wang. – Bristol, 2018. – P. 15–24.
12. Path Planning with Modified a Star Algorithm for a Mobile Robot / F.Duchoň, A. Babinec, M. Kajan, P. Beňo // Procedia Engineering / F.Duchoň, A. Babinec, M. Kajan, P. Beňo. – Amsterdam, 2014. – P. 96.
13. Lentin J. Learning Robotics using Python: Design, simulate, program, and prototype an autonomous mobile robot using ROS, OpenCV, PCL, and Python / Joseph Lentin. – Birmingham: Packt Publishing, 2018. – 280 p.
14. Waypoint Graph Based Fast Pathfinding in Dynamic Environment / W.Zhu, D. Jia, H. Wan, T. Yang // International Journal of Distributed Sensor Networks / W.Zhu, D. Jia, H. Wan, T. Yang. – London, 2015. – P. 1–12.
15. Planar Waypoint Generation and Path Finding in Dynamic Environment / D. Jia, C. Hu, K. Qin, X. Cui // Proceedings of International Conference on Identification, Information and Knowledge in the Internet of Things / D.Jia, C. Hu, K. Qin, X. Cui. – Beijing, 2014. – P. 11–15.
16. Wardhana N. M. Enhanced Waypoint Graph for Path Planning in Virtual Worlds / N. M. Wardhana, H. Johan, H. S. Seah // Proceedings of International Conference on Cyberworlds / N. M. Wardhana, H. Johan, H. S. Seah. – Darmstadt, 2012. – P. 69–76.

17. What is ROS? URL: http://wiki.ros.org/ROS/Introduction (дата звернення: 19.11.2021).
18. Why Gazebo? URL: http://gazebosim.org/ (дата звернення: 19.11.2021).
19. ROS - Data display with Rviz. URL: https://www.stereolabs.com/docs/ros/rviz/ (дата звернення: 19.11.2021).
20. Розроблення стартап-проекту [Електронний ресурс]: Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ«КПІ», 2016. – 28 с.
21. Лавренов Р. О. Математическое и программное обеспечение решения задачи многокритериального поиска пути мобильного объекта / Роман Олегович Лавренов. – Казань: ФГФОУВО “Казанский (Приволжский) федеральный университет”, 2020. – 138 с.
22. Оптимизация расходов на IT. URL: https://www.cfin.ru/itm/tco.shtml (дата звернення: 21.11.2021).
23. Совокупная стоимость владения – современный метод оценки экономической эффективности использования оборудования (на примере конвейерных лент). URL: https://mining-media.ru/ru/article/ekonomic/3572-sovokupnaya-stoimost-vladeniya-sovremennyj-metod-otsenki-ekonomicheskoj-effektivnosti-ispolzovaniya-oborudovaniya-na-primere-konvejernykh-lent (дата звернення: 21.11.2021).
24. Medvecká I. Use of TCO Analysis in Industry 4.0 / I. Medvecká, M. Barbusova, M. Gašo // Industrial Engineering - Invention for Enterprise / I. Medvecká, M. Barbusova, M. Gašo. – Bielsku-Białej, 2019. – P. 28–32.
25. Total Cost of Ownership. URL: https://www.wallstreetmojo.com/total-cost-of-ownership/ (дата звернення: 21.11.2021).
26. The Hidden Costs of Autonomous Mobile Robots – And How to Avoid Them. URL: https://www.roboticsbusinessreview.com/opinion/the-hidden-costs-of-autonomous-mobile-robots-and-how-to-avoid-them/ (дата звернення: 21.11.2021).

27. What does tf do? URL: http://wiki.ros.org/tf (дата звернення: 22.11.2021).
28. Gmapping. URL: http://wiki.ros.org/gmapping (дата звернення: 22.11.2021).
29. Amcl. URL: http://wiki.ros.org/amcl (дата звернення: 24.11.2021).
30. move_base. URL: http://wiki.ros.org/move_base (дата звернення: 25.11.2021).
31. Власов С.М., Бойков В.И., Быстров С.В., Григорьев В.В. Бесконтактные средства локальной ориентации роботов: [Учебное пособие] – Санкт-Петербург: Университет ИТМО, 2017. – 169 с. – экз.
32. Mobile Robots Navigation, Mapping, and Localization Part II (Artificial Intelligence). URL: http://what-when-how.com/artificial-intelligence/mobile-robots-navigation-mapping-and-localization-part-ii-artificial-intelligence/ (дата звернення: 25.11.2021).
33. Скільки людей залишаться без роботи через роботів до 2030 року. URL: https://www.slovoidilo.ua/2018/02/20/infografika/suspilstvo/skilky-lyudej-zalyshatsya-roboty-cherez-robotiv-2030-roku (дата звернення: 01.12.2021).
34. Робота без помилок: приклади роботизації складів у світі та Україні. URL: https://wareteka.com.ua/uk/blog/prikladi-robotizacii-skladiv-u-sviti-ta-ukrayini/ (дата звернення: 05.12.2021).
35. Державна служба статистики України. URL: http://ukrstat.gov.ua/ (дата звернення: 05.12.2021).

# ДОДАТОК А

# ДОДАТОК Б

# ДОДАТОК В

# ДОДАТОК Г

# ДОДАТОК Д

# ДОДАТОК Е

# ДОДАТОК Ж

# ДОДАТОК И

# ДОДАТОК К